

Testeo

Aplicació de gestió d'exàmens tipus tests

UOC

Jordi Cruz Lladó

Màster Universitari en Desenvolupament
de llocs i aplicacions web
Desenvolupament web

Tutor/a de TF

Ignasi Lorente Puchades

**Professor/a responsable de
l'assignatura**

Cesar Pablo Corcoles Briongos

06/2024

Universitat Oberta
de Catalunya

© (Jordi Cruz Lladó)

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel·lectual.

FITXA DEL TREBALL FINAL

Títol del treball:	<i>Testeo – Aplicació de gestió d'exàmens tipus test</i>
Nom de l'autor:	<i>Jordi Cruz Lladó</i>
Nom del consultor/a:	<i>Ignasi Lorente Puchades</i>
Nom del PRA:	<i>Cesar Pablo Corcoles Briongos</i>
Data de lliurament (mm/aaaa):	<i>06/2024</i>
Titulació o programa:	<i>Màster Universitari en Desenvolupament de llocs i aplicacions web</i>
Àrea del Treball Final:	<i>Desenvolupament web</i>
Idioma del treball:	<i>Català</i>
Paraules clau	<i>proves, test, web</i>
Resum del Treball	
<p>L'aplicació de Testeo sorgeix arran de la meua pròpia necessitat de tenir una eina per poder preparar-me els exàmens d'oposició de manera divertida i interactiva. En veure que no hi havia res semblant a la idea que tenia, vaig decidir posar-la en marxa.</p> <p>Testeo és una aplicació web que permet a l'usuari crear els seus exàmens de tipus test per, posteriorment, realitzar-los i ser avaluat. D'aquesta manera l'usuari estudia el temari a l'hora d'introduir les preguntes, i finalment ho valida amb la seva posterior avaluació.</p> <p>La metodologia utilitzada en el projecte ha estat agile, i s'ha utilitzat un sistema de Kanban per planificar la feina.</p> <p>L'aplicació es troba desenvolupada, de manera satisfactòria, amb Spring Boot pel backend i amb Angular i Ionic Framework per al frontend.</p>	
Abstract	
<p>The Testeo application arises from my own need for a tool to prepare for competitive exams in a fun and interactive way. Seeing that there was nothing similar to the idea I had, I decided to bring it to life.</p> <p>Testeo is a web application that allows users to create their own multiple-choice exams, which they can then take and be evaluated on. This way, users study the material while entering the questions and validate their knowledge through subsequent evaluation.</p> <p>The methodology used in the project has been agile, and a Kanban system was employed to plan the work.</p>	

The application is developed with Spring Boot for the backend and Angular and Ionic Framework for the frontend.

Vull dedicar aquest projecte a la Carme, la meva inspiració i suport incondicional en aquest projecte i en tots els aspectes de la vida.

També dedicar-lo als meus pares, per haver-me animat sempre amb el meu somni de dedicar-me a la informàtica.

Moltes gràcies a tots.

Índex

1. Introducció	1
1.1. Context i justificació del Treball	1
1.2. Objectius del Treball	3
1.3. Impacte en sostenibilitat, ètic-social i de diversitat	4
1.4. Enfocament i mètode seguit	6
1.5. Planificació del Treball	6
1.6. Breu sumari de productes obtinguts	9
1.7. Breu descripció dels altres capítols de la memòria	10
2. Disseny centrat en l'usuari	11
2.1. Usuaris i context d'ús	11
2.2. Disseny conceptual	16
2.3. Prototipat	17
3. Disseny tècnic	28
3.1. Arquitectura del sistema	28
3.2. Casos d'ús	28
3.3. Base de dades	37
4. Disseny tècnic	38
4.1. Decisions de disseny	38
4.2. Llibreries utilitzades	38
4.3. Desenvolupament	39
4.4. Punts de control	42
4.5. Errors del programari	44
4.6. Execució del codi	44
5. Conclusions i treball futur	46
5.1. Conclusions del treball	46
5.2. Treball futur	47
6. Glossari	48
7. Bibliografia	51
8. Annexos	53

Llista de figures

Fig. 1. Comparativa de consum d'energia per l'algorisme de les N-reines [6]. ...	5
Fig. 2. Diagrama de Gantt de la planificació	7
Fig. 3. Participació per perfil.....	13
Fig. 4. Respostes rellevants perfil TIC	13
Fig. 5. Respostes rellevants perfil administratiu	14
Fig. 6. Respostes rellevants perfil estudiant.....	15
Fig. 7. Respostes generals de la part final	15
Fig. 8. Arbore de navegació de l'aplicació	16
Fig. 9. Onboarding (<i>wireframes</i>)	17
Fig. 10. Inici de sessió d'usuari i registre d'usuari (<i>wireframes</i>).....	18
Fig. 11. Llista de test. Buida i amb dades (<i>wireframes</i>)	18
Fig. 12. Detall d'un test. Dades generals, confirmació d'eliminar test i confirmació d'eliminar pregunta (<i>wireframes</i>).....	19
Fig. 13. Creació/modificació de test. Dades generals i de la pregunta (<i>wireframes</i>)	19
Fig. 14. Realització i revisió del test (<i>wireframes</i>)	20
Fig. 15. Favorits. Buits i amb dades (<i>wireframes</i>)	20
Fig. 16. Perfil d'usuari (<i>wireframes</i>)	21
Fig. 17. Onboarding (<i>mockups</i>)	21
Fig. 18. Inici de sessió d'usuari i registre d'usuari (<i>mockups</i>).....	22
Fig. 19. Llista de test. Buida i amb dades (<i>wireframes</i>)	22
Fig. 20. Detall d'un test. Dades generals, confirmació d'eliminar test i confirmació d'eliminar test i pregunta (<i>mockups</i>).....	23
Fig. 21. Creació/modificació de test. Dades generals i de la pregunta (<i>mockups</i>)	23
Fig. 22. Realització i revisió del test (<i>mockups</i>)	24
Fig. 23. Favorits. Buits i amb dades (<i>mockups</i>)	24
Fig. 24. Perfil d'usuari (<i>mockups</i>).....	25
Fig. 25. Icona de l'aplicació	25
Fig. 26. Logotip de l'aplicació.....	25
Fig. 27. Tecnologies utilitzades.....	28
Fig. 28. Diagrama UML dels casos d'ús	29
Fig. 29. Diagrama Entitat-Relació de l'aplicació	37

Llista de taules

Taula I - Planificació temporal per fites.....	6
Taula II - Planificació de la definició del projecte	8
Taula III - Planificació del disseny i primera versió	8
Taula IV - Planificació de desenvolupament del producte	9
Taula V - Planificació de lliurament del producte	9
Taula VI - Enquesta d'usuaris	12
Taula VII - Colors de l'aplicació.....	26
Taula VIII - Onboarding (cas d'ús)	30
Taula IX - Registrar-se (cas d'ús)	30
Taula X - Login (cas d'ús)	31
Taula XI - Consultar llista de tests (cas d'ús)	31

Taula XII - Crear un test (cas d'ús)	31
Taula XIII - Consultar dades d'un test (cas d'ús).....	32
Taula XIV - Editar les dades d'un test (cas d'ús).....	32
Taula XV - Editar una pregunta (cas d'ús)	33
Taula XVI - Afegir una pregunta (cas d'ús).....	33
Taula XVII - Eliminar una pregunta (cas d'ús).....	34
Taula XVIII - Eliminar un test (cas d'ús)	34
Taula XIX - Afegir un test a favorits (cas d'ús)	35
Taula XX - Realitzar un test (cas d'ús)	35
Taula XXI - Consultar llista de favorits (cas d'ús).....	36
Taula XXII - Consultar dades de sessió (cas d'ús)	36
Taula XXIII - Tancar sessió (cas d'ús)	36

1. Introducció

1.1. Context i justificació del Treball

Aquest TFM sorgeix d'una necessitat que he tingut durant un any i intenta resoldre de cara a futures opcions o ajudar a persones que es puguin trobar en una situació semblant a la meua.

Durant el darrer any he estat estudiant per preparar unes oposicions per funcionari a la UIB. Aquestes oposicions consistien en un examen tipus test de setanta preguntes, de quatre respostes amb una sola correcta, on cada pregunta errònia restava una cinquena part d'una pregunta correcta. Tot i que per estudiar aquests tipus de prova el convenient és llegir al peu de la lletra el temari, és una bona opció també poder constituir els teus propis exàmens tipus test per dos motius: per repassar el contingut a mesura que elabores l'examen; i per posar a prova els teus coneixements.

Al llarg d'aquest darrer any vaig cercar si existia alguna aplicació que em permetés generar i realitzar aquests tipus de prova de manera senzilla, gratuïta i, sobretot, multiplataforma, però cap satisfia aquests tres principis.

Anem a veure cinc de les aplicacions més destacades que podem trobar actualment com a competidores d'aquesta que desenvoluparem i la seva justificació per cada un dels principis anteriors:

Google Forms

Aplicació arxivada per tothom, desenvolupada per Google. És una eina que permet definir enquestes, qüestionaris i formularis personalitzats, així com establir respostes simples, múltiples, etc. També permet visualitzar en temps real els resultats i obtenir *feedback* davant respostes incorrectes, i obtenir informes i estadístiques [1].

Valoració dels principis:

- **Senzilla:** l'aplicació té moltes opcions per realitzar diversos tipus de formularis i, tot i que és bastant intuïtiva, ofereix massa opcions que poden confondre a l'usuari que vol realitzar només proves de tipus tests.
- **Gratuïta:** té una modalitat gratuïta com a part de Google Workspace, que no limita respecte a funcionalitat.
- **Multiplataforma:** disposa tant d'aplicació web com d'aplicació per mòbil.

Aquesta aplicació sembla que podria encaixar davant la necessitat principal del projecte, però té un principal punt en contra: no permet establir regles per restar puntuació quan es falla una pregunta, tot i que és possible aconseguir-ho amb scripts. A més, volem una eina especialitzada en tipus test, que no ens puguin distreure les altres eines per fer altre tipus de formulari.

Kahoot!

Aplicació utilitzada en l'àmbit acadèmic, entre altres, per crear qüestionaris, enquestes i jocs de preguntes i respostes interactives. Si bé no permet establir una nota numèrica, tal com es desitja a un examen tipus test, atorga una puntuació als usuaris participants [2].

Valoració dels principis:

- **Senzilla:** la interfície es troba massa sobrecarregada d'opcions i de publicitat pròpia de diferents continguts que ofereixen, i la creació de preguntes és molt limitada en la versió gratuïta.
- **Gratuïta:** té una modalitat gratuïta que permet crear preguntes de quatre respostes o de vertader i fals. La resta de contingut és de pagament.
- **Multiplataforma:** disposa tant d'aplicació web com d'aplicació per mòbil.

Aquesta aplicació no encaixa gaire al que perseguim, ja que els exàmens d'oposicions poden variar en quantitat de respostes, i tampoc ens permet rebre un *feedback* ni obtenir una qualificació numèrica com si fos un examen. També, en el pla gratuït, es troba molt limitada d'opcions. El millor que té és la *gamificació* i la diversió a l'hora de fer partícips als usuaris.

Socrative

Aplicació utilitzada principalment en l'àmbit acadèmic per crear qüestionaris, avaluacions i activitats interactives en temps real. Permet obtenir *feedback* davant les respostes de l'usuari, obtenir informes i estadístiques, així com obtenir una nota numèrica com si es tractés d'un examen. Important: permet definir un sistema de penalització pel qual les preguntes incorrectes restin puntuació [3].

Valoració dels principis:

- **Senzilla:** la interfície es senzilla i agradable d'utilitzar. Té diferents opcions per configurar les diferents proves, així com per obtenir estadístiques.
- **Gratuïta:** té una modalitat gratuïta que permet crear un màxim de cinc exàmens. La resta de limitacions no ens afecten.
- **Multiplataforma:** disposa tant d'aplicació web com d'aplicacions per mòbil. En aquesta, es diferencien dues aplicacions: Socrative Teacher, que permet definir els tests; Socrative Student, per dur a terme els tests.

Si bé aquesta aplicació té molta funcionalitat de què volem disposar té una limitació molt important en la seva versió gratuïta: com a màxim cinc exàmens. Per unes oposicions és inqüestionable la necessitat de poder disposar de més exàmens. També, haver de menester dues aplicacions no acaba d'encaixar dins el concepte de multiplataforma que cerquem.

Quizlet

Aquesta aplicació va néixer amb l'objectiu d'aprendre vocabulari en altres idiomes gràcies al sistema de *flashcards*, però ha anat evolucionant fins al punt d'oferir diferents tipus de proves, com ara tipus test [4]. Tot això, Quizlet no es troba enfocada a una avaluació de tipus examen, ja que no proveeix de nota numèrica un cop acabada la prova.

Valoració dels principis:

- **Senzilla:** la interfície es senzilla i moderna, i proporciona el necessari d'una sola vegada per definir les preguntes, encara que té poques opcions per configurar les proves.
- **Gratuïta:** té una modalitat gratuïta que no limita en quantitat, però sí en opcions. A més, apareix publicitat.
- **Multiplataforma:** disposa tant d'aplicació web com d'aplicació per mòbil.

El millor que té és la interfície, perquè quant a funcionalitat i opcions és molt limitada i no satisfà l'objectiu que es persegueix. Sobretot, perquè no proporciona una nota numèrica i, per tant, tampoc pot establir un sistema de penalització de resposta incorrecta.

Classmarker

Aplicació amb l'objectiu de permetre crear, administrar i avaluar exàmens en línia. Proporciona diferents configuracions de les proves, com resposta múltiple, temps límit, etc. Permet obtenir una nota numèrica i, important, definir un sistema de penalització pel qual les preguntes incorrectes restin puntuació [5].

Valoració dels principis:

- **Senzilla:** la interfície és senzilla però genèrica, ja que sembla una mica desactualitzada, amb estils antiquats.
- **Gratuïta:** té una modalitat gratuïta que limita a 100 tests per mes, redueix funcionalitat i, a més, apareix publicitat.
- **Multiplataforma:** disposa només d'aplicació web.

En relació amb la funcionalitat aquesta podria encaixar, però no ho fa en relació amb la senzillesa i multiplataforma. Es cerca que sigui de fàcil configuració i la interfície és complexa, però el més greu és que només disposa d'aplicació web.

1.2. Objectius del Treball

En la realització d'aquest Treball de Fi de Màster hi ha dos tipus diferenciats d'objectius que es volen assolir. Per una banda, els objectius principals, que són els que defineixen el que es pretén aconseguir amb aquest treball, i, per altra banda, els objectius secundaris, que són els que complementen i enriqueixen l'abast del projecte, fomentant el desenvolupament de competències addicionals.

Els objectius principals són els següents:

- Desenvolupar una aplicació web que permeti als usuaris gestionar exàmens de tipus test, amb la seva creació, modificació i configuració de l'examen i el seu contingut.
- Aplicar els coneixements adquirits en el Màster de Desenvolupament de llocs i aplicacions web.
- Implementar funcionalitat perquè els usuaris puguin realitzar els exàmens de tipus tests, ser avaluats i obtenir *feedback* davant les respostes.
- Implementar un mòdul d'autenticació i registre d'usuaris per protegir l'aplicació, i fer que cada usuari disposi dels seus exàmens.
- Dissenyar una interfície d'usuari intuïtiva i amigable per garantir una experiència d'usuari satisfactòria.

Els objectius secundaris són els següents:

- Aprendre l'ús del *framework* Ionic per generar components que semblin natius segons el dispositiu d'accés a l'aplicació.
- Millorar les competències en Angular apreses durant la realització d'aquest màster.
- Aprendre a treballar juntament amb Angular y Spring Boot, prescindint de les plantilles *JSP* o *Thymeleaf*.

A part dels objectius del TFM, hem d'obtenir i identificar els requeriments que ha de satisfer l'aplicació web que durem a terme, de manera que tindrem una descripció clara i detallada del que ha de fer i com, i també servirà per poder validar que tota la funcionalitat ha estat coberta en la fase de finalització del projecte:

Requeriments funcionals:

- El sistema ha de permetre l'enregistrament d'usuari per poder accedir-hi.
- El sistema ha de permetre l'autenticació de l'usuari per accedir al seu contingut.
- El sistema ha de permetre a l'usuari crear, definir i parametritzar un examen de tipus test.
- El sistema ha de permetre a l'usuari categoritzar un examen de tipus test.
- El sistema ha de permetre a l'usuari realitzar l'examen i obtenir una puntuació.
- El sistema ha de permetre a l'usuari consultar tots els exàmens dels qual disposa.
- El sistema ha de permetre a l'usuari consultar l'històric de qualificacions obtingudes per un examen.

Requeriments no funcionals:

- El sistema, inicialment, ha d'estar localitzat en l'idioma català.
- El sistema només ha d'acceptar peticions HTTPs per la seguretat de la informació en trànsit.
- La comunicació entre el *backend* i el *frontend* es realitzarà mitjançant un *token* únic per usuari.
- El sistema tindrà una interfície que s'adaptarà al dispositiu des d'on s'accedeixi, mostrant els components com si fossin natius.
- El *backend* del sistema ha d'estar desenvolupat amb Java 17 i Spring Boot.
- El *frontend* del sistema ha d'estar desenvolupat amb Angular i Ionic Framework.
- El sistema ha d'informar a l'usuari dels errors que hi pugui haver de manera senzilla i amigable.

1.3. Impacte en sostenibilitat, ètic-social i de diversitat

El desenvolupament d'aquesta aplicació té un impacte positiu sobre l'ODS 12, producció i consum responsable, ja que l'aplicació contribueix a la reducció de l'ús de paper i altres recursos materials per la generació d'exàmens de tipus tests, resultant en un menor consum de recursos naturals i menor generació de residus.

Tot això, podria tenir també un efecte negatiu al mateix ODS, ja que l'aplicació necessita recursos per poder executar-se i, si aquests no es troben optimitzats i mal gestionats, podria dur un excés de consum de recursos. Per això, la lògica de l'aplicació es desenvoluparà emprant el llenguatge de programació Java, ja que és un dels llenguatges amb millor empremta verda segons diferents proves de rendiment [6].

Energy		Time		Mb	
(c) C	1.00	(c) C	1.00	(c) Pascal	1.00
(c) Rust	1.03	(c) Rust	1.04	(c) Go	1.05
(c) C++	1.34	(c) C++	1.56	(c) C	1.17
(c) Ada	1.70	(c) Ada	1.85	(c) Fortran	1.24
(v) Java	1.98	(v) Java	1.89	(c) C++	1.34
(c) Pascal	2.14	(c) Chapel	2.14	(c) Ada	1.47
(c) Chapel	2.18	(c) Go	2.83	(c) Rust	1.54
(v) Lisp	2.27	(c) Pascal	3.02	(v) Lisp	1.92
(c) Ocaml	2.40	(c) Ocaml	3.09	(c) Haskell	2.45
(c) Fortran	2.52	(v) C#	3.14	(i) PHP	2.57
(c) Swift	2.79	(v) Lisp	3.40	(c) Swift	2.71
(c) Haskell	3.10	(c) Haskell	3.55	(i) Python	2.80
(v) C#	3.14	(c) Swift	4.20	(c) Ocaml	2.82
(c) Go	3.23	(c) Fortran	4.20	(v) C#	2.85
(i) Dart	3.83	(v) F#	6.30	(i) Hack	3.34
(v) F#	4.13	(i) JavaScript	6.52	(v) Racket	3.52
(i) JavaScript	4.45	(i) Dart	6.67	(i) Ruby	3.97
(v) Racket	7.91	(v) Racket	11.27	(c) Chapel	4.00
(i) TypeScript	21.50	(i) Hack	26.99	(v) F#	4.25
(i) Hack	24.02	(i) PHP	27.64	(i) JavaScript	4.59
(i) PHP	29.30	(v) Erlang	36.71	(i) TypeScript	4.69
(v) Erlang	42.23	(i) Jruby	43.44	(v) Java	6.01
(i) Lua	45.98	(i) TypeScript	46.20	(i) Perl	6.62
(i) Jruby	46.54	(i) Ruby	59.34	(i) Lua	6.72
(i) Ruby	69.91	(i) Perl	65.79	(v) Erlang	7.20
(i) Python	75.88	(i) Python	71.90	(i) Dart	8.64
(i) Perl	79.58	(i) Lua	82.91	(i) Jruby	19.84

Fig. 1. Comparativa de consum d'energia per l'algorisme de les N-reines [6].

A causa de la naturalesa i funcionalitat específiques de l'aplicació de prova tipus test, el seu abast es limita a la gestió i administració d'exàmens de tipus test. Tampoc implica interaccions socials complexes ni activitats que puguin causar dilemes ètics o un impacte social significatiu.

En termes de responsabilitat social, l'aplicació té com a objectiu brindar una eina útil per al sector educatiu, permetent a qualsevol persona crear i fer avaluacions de manera efectiva i eficient. Pel fet que aquesta aplicació s'usa principalment en un context educatiu, s'espera que els usuaris utilitzin aquesta aplicació de manera responsable i ètica.

No obstant això, qualsevol impacte ètic i social negatiu que pugui sorgir de l'ús de l'aplicació es relacionarà amb el comportament dels usuaris individuals i no amb l'aplicació en si.

Aquest TFM s'enfoca principalment en aspectes tècnics relacionats amb la gestió i administració de proves d'avaluació. Donada aquesta naturalesa específica, el seu impacte directe en aspectes de diversitat, gènere i drets humans és limitat.

L'aplicació no recopila ni processa dades que estiguin directament relacionats amb aquestes dimensions socials, i no discrimina ni afavoreix a cap grup específic en funció de la seva identitat de gènere, raça, religió o altres característiques protegides.

1.4. Enfocament i mètode seguit

Aquesta aplicació no parteix de cap aplicació existent ni de cap enfocament previ, ja que es pretén obtenir una aplicació especialitzada en la creació d'exàmens de tipus test a partir d'un desenvolupament propi, des de zero, atès que les eines de l'estudi de mercat no són de codi lliure i, per tant, no es pot fer cap millora d'aquestes. També, fent el desenvolupament propi, posem en pràctica i demostrem l'assoliment de les competències adquirides en el màster.

Es decideix fer el *backend* en Spring Boot per l'experiència prèvia en aquesta tecnologia, per poder desenvolupar àgilment la lògica de negoci de l'aplicació. El *frontend* es desenvoluparà en Angular, tal i com hem après en el màster, però incorporant el *framework* Ionic per millorar l'experiència d'usuari i per aprendre una tecnologia més. Pel que fa a la base de dades, s'utilitzarà MariaDB ja que és una aplicació senzilla i continguda, i no necessitem la flexibilitat o capacitat de maneig de grans quantitats de dades que ens pot aportar una base de dades de tipus NoSQL. El codi es trobarà a un repositori de GitHub, i durant el desenvolupament s'utilitzarà com a control de versions.

Pel que fa a la metodologia de desenvolupament, al tractar-se d'un desenvolupament unipersonal, el més adient és utilitzar el mètode Kanban, ja que no té sentit utilitzar Scrum al ser un marc de feina orientat a petits equips [7], ni tampoc un desenvolupament en cascada, ja que aquesta aplicació pot anar sufrint canvis durant el seu desenvolupament, i també volem poder tenir prototipats des de l'inici del desenvolupament [8]. Kanban ens permet optimitzar el flux de treball en un procés o sistema, sent el seu principal objectiu maximitzar l'eficiència i minimitzar el temps de lliurament en eliminar el desaprofitament i optimitzar el flux de treball [9].

1.5. Planificació del Treball

Per planificar el treball per fer hem tengut en compte les obligacions inexcusables, tant laborals com familiars. Per això, ens surt una mitjana de dedicació de 3 hores diàries, caps de setmana i festius inclosos. Agafant dia 28 de febrer com a data inicial i dia 16 de juny com a data final, disposam de 110 dies per fer el TFM, dels quals correspondran 330 hores de dedicació (superant per 30 hores les 300 que correspondrien als 12 crèdits ECTS del TFM, les quals tindrem com a contingència per imprevistos).

En la següent taula mostrarem la dedicació desglossada segons cada data de lliurament de les PAC:

Taula I - Planificació temporal per fites

Fita	Data d'inici	Data de fi	Hores
PAC1 – Definició del projecte	28/02/2024	12/03/2024	42
PAC2 – Disseny i primera versió	13/03/2024	14/04/2024	99
PAC3 – Desenvolupament del producte	15/04/2024	19/05/2024	105
PAC4 – Lliurament del producte	20/05/2024	16/06/2024	84
Total			330

A continuació, anem a elaborar una planificació temporal per cada una de les fites de la taula anterior, de manera que ens permeti detectar si anem sobre la planificació, o bé hi ha desviaments temporals i hem de fer qualche correcció per poder lliurar a temps.

També aportarem un diagrama de Gantt per a visualitzar de manera clara i estructurada el calendari de tasques i el seu cronograma d'execució al llarg del projecte. Això ajudarà a gestionar el temps i els recursos de manera eficient i a mantenir un seguiment adequat del progrés del projecte.

Diagrama de Gantt

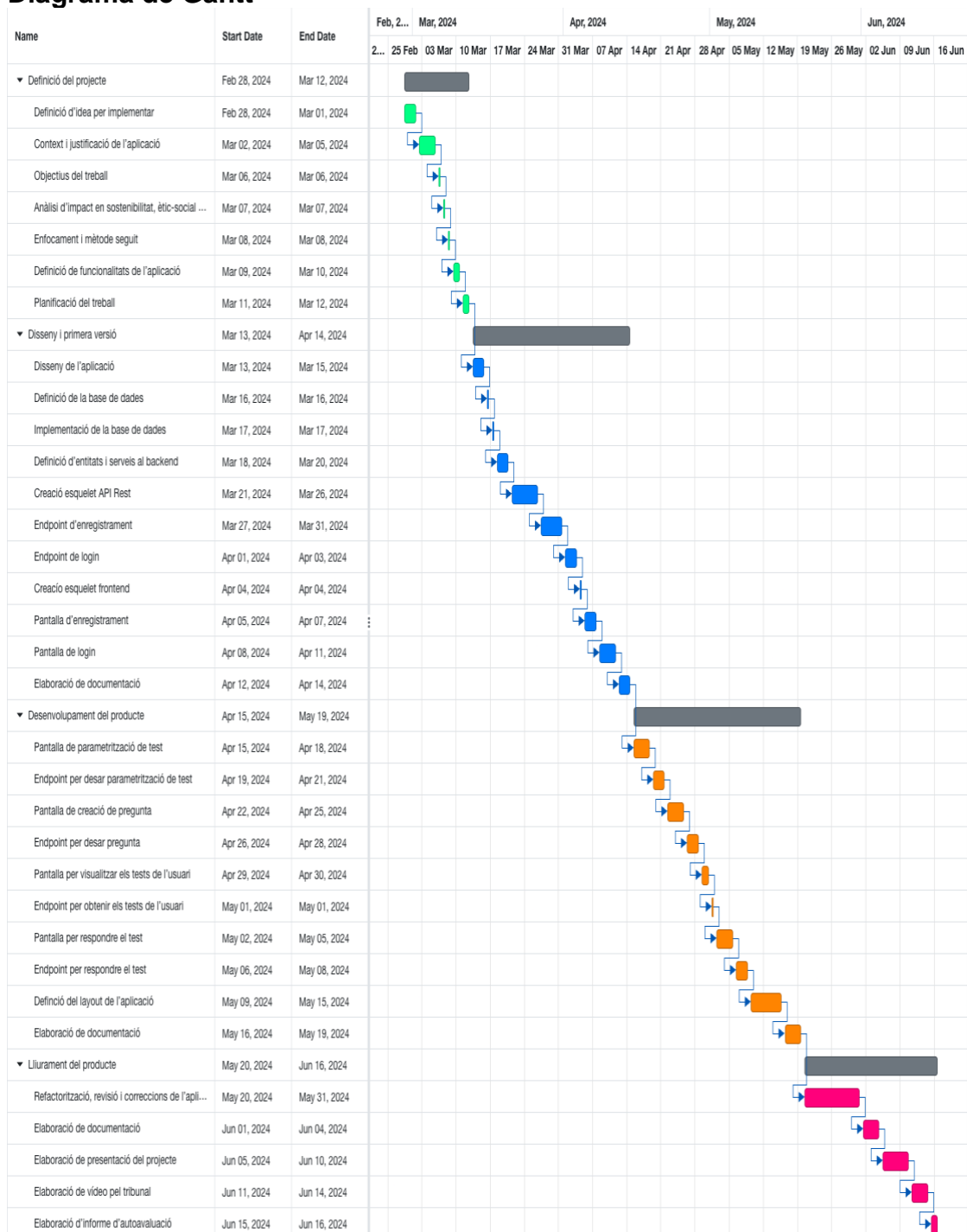


Fig. 2. Diagrama de Gantt de la planificació

Definició del projecte

En aquesta etapa, s'estableix una visió general del projecte, definint els objectius que es volen aconseguir, detallant el context d'ús a través d'una anàlisi del mercat i realitzant una planificació temporal detallada que inclou les tasques a realitzar durant tot el projecte.

Taula II - Planificació de la definició del projecte

Tasca	Data d'inici	Data de fi	Hores
P1.1 - Definició d'idea per implementar	28/02/2024	01/03/2024	9
P1.2 - Context i justificació de l'aplicació	02/03/2024	05/03/2024	12
P1.3 - Objectius del treball	06/03/2024	06/03/2024	3
P1.4 - Anàlisi d'impacte en sostenibilitat, ètic-social i de diversitat	07/03/2024	07/03/2024	3
P1.5 - Enfocament i mètode seguit	08/03/2024	08/03/2024	3
P1.6 - Definició de funcionalitats de l'aplicació	09/03/2024	10/03/2024	6
P1.7 - Planificació del treball	11/03/2024	12/03/2024	6
Total			42

Disseny i primera versió

En aquesta etapa, es comença a dissenyar i implementar els fonaments de l'aplicació, com ara el disseny o la base de dades on emmagatzemar totes les dades. També inclou la revisió i modificació de la documentació del TFM en relació amb les tasques fetes.

Taula III - Planificació del disseny i primera versió

Tasca	Data d'inici	Data de fi	Hores
P2.1 - Disseny de l'aplicació	13/03/2024	15/03/2024	9
P2.2 - Definició de la base de dades	16/03/2024	16/03/2024	3
P2.3 - Implementació de la base de dades	17/03/2024	17/03/2024	3
P2.4 - Definició d'entitats i serveis al <i>backend</i>	18/03/2024	20/03/2024	9
P2.5 - Creació esquelet API Rest	21/03/2024	26/03/2024	18
P2.6 - Endpoint d'enregistrament	27/03/2024	31/03/2024	15
P2.7 - Endpoint de login	01/04/2024	03/04/2024	9
P2.8 - Creació esquelet <i>frontend</i>	04/04/2024	04/04/2024	3
P2.9 - Pantalla d'enregistrament	05/04/2024	07/04/2024	9
P2.10 - Pantalla de login	08/04/2024	11/04/2024	12
P2.11 - Elaboració de documentació	12/04/2024	14/04/2024	9
Total			99

Desenvolupament del producte

En aquesta etapa, s'implementa el gruix de l'aplicació. És a dir, es realitza la implementació dels requeriments que ha de satisfer l'aplicació. També inclou la revisió i modificació de la documentació del TFM en relació amb les tasques fetes.

Taula IV - Planificació de desenvolupament del producte

Tasca	Data d'inici	Data de fi	Hores
P3.1 - Pantalla de parametrització de test	15/04/2024	18/04/2024	12
P3.2 - Endpoint per desar parametrització de test	19/04/2024	21/04/2024	9
P3.3 - Pantalla de creació de pregunta	22/04/2024	25/04/2024	12
P3.4 - Endpoint per desar pregunta	26/04/2024	28/04/2024	9
P3.5 - Pantalla per visualitzar els tests de l'usuari	29/04/2024	30/04/2024	6
P3.6 - Endpoint per obtenir els tests de l'usuari	01/05/2024	01/05/2024	3
P3.7 - Pantalla per respondre el test	02/05/2024	05/05/2024	12
P3.8 - Endpoint per respondre el test	06/05/2024	08/05/2024	9
P3.9 - Definició del layout de l'aplicació	09/05/2024	15/05/2024	21
P3.10 - Elaboració de documentació i vídeo	16/05/2024	19/05/2024	12
Total			105

Lliurament del producte

Etapa final del desenvolupament, on es durà a terme una anàlisi del codi desenvolupat pel seu refinament, així com la correcció d'errors que hi puguin haver. També, es desenvoluparà la documentació i recursos necessaris per a l'entrega efectiva del TFM.

Taula V - Planificació de lliurament del producte

Tasca	Data inici	Data fi	H.
P4.1 - Refactorització, revisió i correccions de l'aplicació	20/05/2024	31/05/2024	36
P4.2 - Elaboració de documentació	01/06/2024	04/06/2024	12
P4.3 - Elaboració de presentació del projecte	05/06/2024	10/06/2024	18
P4.4 - Elaboració de vídeo pel tribunal	11/06/2024	14/03/2024	12
P4.5 - Elaboració d'informe d'autoavaluació	15/03/2024	16/06/2024	6
Total			84

1.6. Breu sumari de productes obtinguts

Juntament amb la memòria s'entregaran els següents productes:

- Codi font de l'aplicació.
- Manual d'instal·lació, compilació i execució.
- Vídeo-presentació del projecte.
- Direcció web pel seu accés.

1.7. Breu descripció dels altres capítols de la memòria

En aquesta memòria trobareu la informació estructurada i classificada en diferents capítols. El segon capítol explica com s'ha realitzat l'estudi i disseny que tindrà l'aplicació mitjançant la filosofia DCU: usuaris i context d'ús, el disseny conceptual de l'aplicació, i els prototips Lo-Fi i Hi-Fi, juntament amb les seves conclusions. Després, al tercer capítol, s'explica el disseny tècnic de l'aplicació, on s'inclouran els diagrames i casos d'ús, així com el disseny de la base de dades. Al quart capítol trobarem el disseny tècnic: decisions preses, llibreries utilitzades en el desenvolupament, errors de programari i com executar el codi. Finalment, en el cinquè capítol trobarem les conclusions obtingudes un cop realitzat el projecte, així com treball futur un cop acabat aquest.

2. Disseny centrat en l'usuari

Per dur a terme el disseny de l'aplicació utilitzarem la filosofia DCU, o Disseny Centrat en l'Usuari. Aquesta DCU és una manera de concebre productes o serveis, on es posa èmfasis en les necessitats i desitjos de l'usuari des de l'inici fins a la culminació del procés de disseny [10].

Aquest canvi de perspectiva implica la participació activa dels usuaris en totes les fases del projecte, permetent una avaluació constant del producte des de diferents punts de vista. Per a aconseguir-ho, el disseny centrat en l'usuari emprà una varietat de mètodes i tècniques de recerca i disseny.

Aquesta filosofia es sol estructurar en aquestes quatre fases:

- **Entendre el context d'ús:** s'identifiquen als usuaris del producte, les seves necessitats i com l'utilitzaran.
- **Especificar els requeriments:** s'identifiquen els requeriments tant de l'empresa (nosaltres) com dels usuaris que l'utilitzaran.
- **Dissenyar les solucions:** les fases pròpies del desenvolupament.
- **Avaluar els resultats:** revisió i validació de que el producte resultant satisfà els requeriments recollits.

2.1. Usuaris i context d'ús

En aquesta fase es recull la informació més rellevant dels possibles usuaris finals de l'aplicació, mitjançant mètodes d'investigació que involucren els usuaris per entendre les seves característiques, necessitats i objectius. En aquest cas, hem optat per utilitzar el mecanisme d'enquestes, ja que són una tècnica quantitativa que permet obtenir informació de manera remota d'una mostra representativa d'usuaris.

Les enquestes es duran a terme a diferents grups de persones amb perfils diferenciats:

- **Perfil TIC:** persones que desenvolupen aplicacions, tant *frontend* com *backend*, que tenen soltesa amb les interfícies d'aplicacions.
- **Perfil administratiu:** persones que en el seu dia a dia empen aplicacions concretes i/o d'ofimàtica, però amb poc domini de les TIC.
- **Perfil estudiant *senior*:** persones que es troben en situació d'estudiant i tenen trenta anys o més. Si fan feina, s'ignora aquesta situació i s'agafa només el perfil d'estudiant.
- **Perfil estudiant *junior*:** persones que es troben en situació d'estudiant i tenen menys de trenta anys. Si fan feina, s'ignora aquesta situació i s'agafa només el perfil d'estudiant.

Es tria el tall d'edat en trenta anys, ja que l'ésser humà assoleix el màxim de la memòria a curt termini als vint-i-cinc anys i la mantén durant una dècada [11]. Per tant, agafem el punt mitjà com a tall d'edat diferenciadora.

Per poder dur a terme l'enquesta de manera telemàtica, s'ha dissenyat amb Google Forms. Aquesta enquesta té una pregunta clau que, basant-se en la resposta elegida per l'usuari, mostrarà una secció diferent. L'enquesta es pot consultar en el següent enllaç: [enquesta d'usuaris](#).

A continuació posarem l'enquesta en format clàssic per facilitar la seva lectura.

Taula VI - Enquesta d'usuaris

Secció general (comuna per tots)
<ul style="list-style-type: none"> - Quina edat tens? - Quin és el teu gènere? - Utilitzes sovint aparells electrònics? - Quina d'aquestes situacions et descriu millor?
Secció TIC
<ul style="list-style-type: none"> - Trobes rellevant que l'aplicació permeti exportar o integrar els exàmens tipus test amb altres plataformes? - Trobes rellevant que l'aplicació sigui open-source a l'hora d'escollir-ne una? - Quin nivell d'importància li dones a la compatibilitat multiplataforma en una aplicació d'exàmens tipus test? - Sobre el disseny d'interfície de les aplicacions...
Secció administratiu
<ul style="list-style-type: none"> - Amb quina freqüència utilitzes aplicacions informàtiques en el teu treball? - Quina consideres que és la característica més important en una aplicació d'exàmens tipus test? - Trobes que pot ser útil per la teva situació una aplicació d'exàmens de tipus test?
Secció estudiant
<ul style="list-style-type: none"> - Quin nivell d'estudis curses actualment? - Amb quina freqüència et trobes amb avaluacions en format test? - Utilitzes alguna eina per estudiar exàmens tipus test? - Quina configuració de feedback t'agradaria rebre en una aplicació d'exàmens de tipus test? - Quina visualització de preguntes t'agradaria més?
Secció final (comuna per tots)
<ul style="list-style-type: none"> - Quin dispositiu utilitzes amb més freqüència? - Estaries disposat/da a pagar per una aplicació d'exàmens tipus test? - En cas que sí, quin model de pagament prefereixes? - Amb quina finalitat utilitzaries una aplicació d'exàmens de tipus test?

Hem comptabilitzat les respostes a dia tres d'abril de 2024, que era la fita que ens havíem marcat per tancar-les. En aquest moment, hi ha **trenta-una enquestes contestades**, on d'aquestes respostes vuit (25,8%) són de perfil TIC, onze (35,5%) de perfil administratiu i dotze (38,7%) de perfil estudiant.

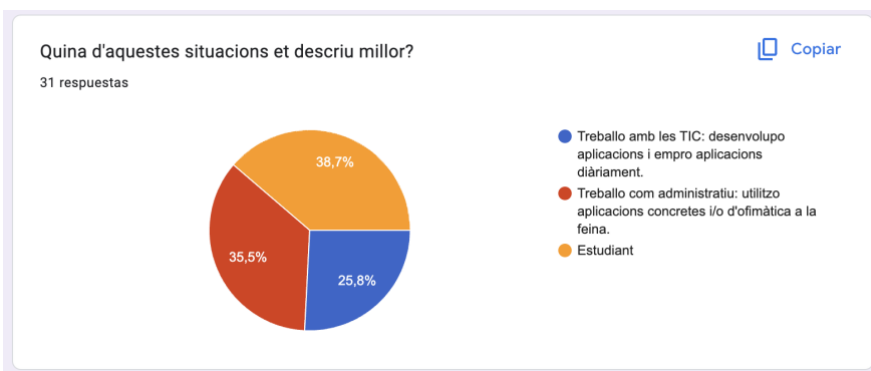


Fig. 3. Participació per perfil

Destacar que només el 12,9% dels usuaris no utilitzen sovint aparells electrònics. Observem que la participació ha estat **quasi paritària en relació amb el gènere**, ja que el 45,2% és del gènere femení, el 48,2% del gènere masculí, i un 6,5% ha preferit no contestar-ho. En relació a la edat, el 51,6% tenen trenta anys o més, i el 48,4% menys de trenta anys.

Ara, si analitzem les respostes de cadascun dels perfils, trobem el següent:

Pel cas de perfil TIC, el 100% troben que **és irrellevant exportar les dades** o que s'integri amb altres aplicacions. Això ens fa descartar incloure aquesta característica. També, el 62,5% **no li dona importància a si l'aplicació és open-source** a l'hora d'escollir-ne una. Això sí, el 62,5% troba que **és molt important que l'aplicació sigui multiplataforma**, seguit del 25% que troben que és important que ho sigui. Aquest resultat recolza que el projecte sigui una aplicació web per poder accedir-hi des de qualsevol dispositiu. Finalment, en relació amb el disseny de la interfície, el 62,5% troba que **ha d'adaptar-se al look-and-feel del sistema operatiu, però aportar un punt que la diferenciï de les altres**. Inicialment, volíem emprar Ionic per tenir un estil 100% del sistema operatiu, però analitzant el resultat inclourem un toc diferenciador.

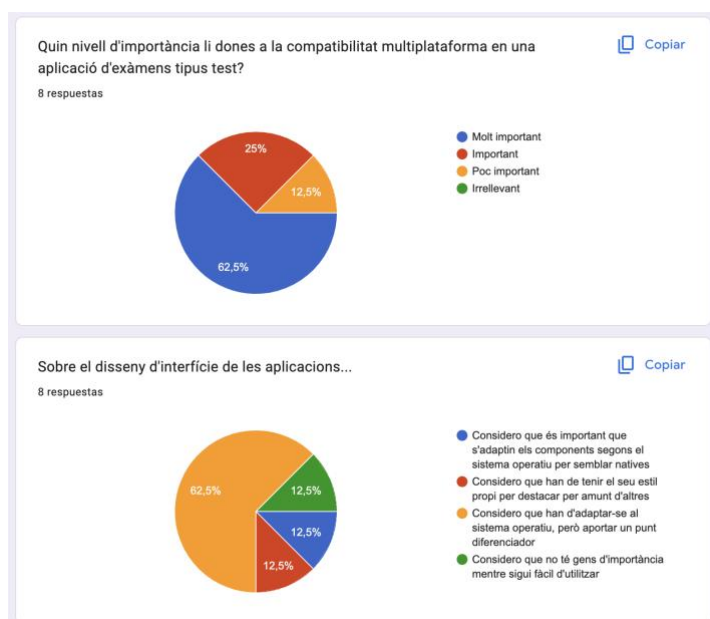


Fig. 4. Respostes rellevants perfil TIC

Pel cas de perfil administratiu, la gran majoria (45,5%) utilitza a diari aplicacions informàtiques en el lloc de feina, seguit d'un 27,3% que les utilitzen entre dues i tres vegades per setmana. Només el 9,1% les utilitza de manera puntual. Amb això aconseguim **tenir una varietat d'usuaris no experts que ens ajudaran a tenir una perspectiva diferent de la nostra professional (TIC)**. El 100% d'ells troben que la **característica més important ha de ser la seva facilitat d'ús**, deixant de banda el nivell de configuració o les estadístiques que es puguin generar. Tot i així, només el 63,5% trobarien útil aquesta aplicació per la seva situació.

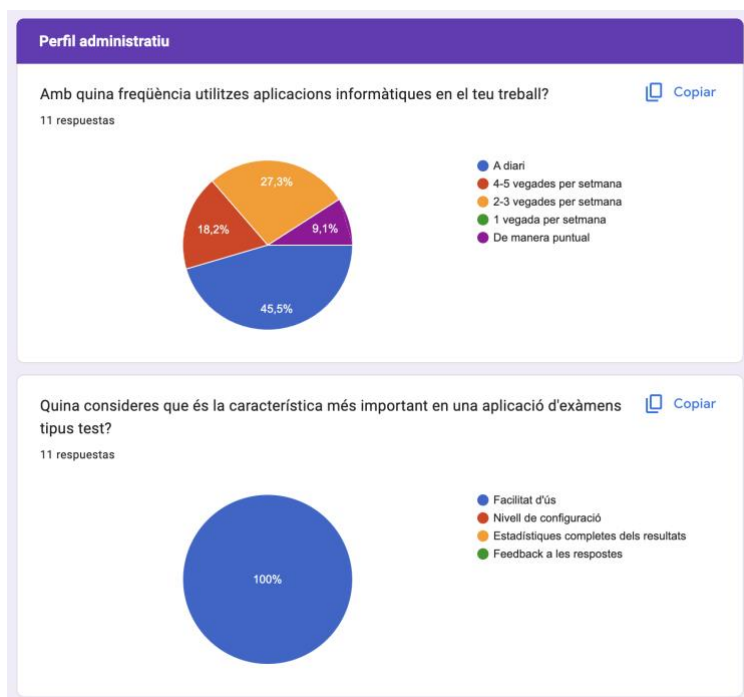


Fig. 5. Respostes rellevants perfil administratiu

Pel cas de perfil estudiant, que és el nostre potencial nínxol, el 83,3% cursen estudis de nivell superior (tant universitaris com formació professional superior), i la resta es reparteix entre graduat escolar i batxiller/formació professional I. De tots ells, **només el 25% no té cap mena d'avaluació de tipus test**, però això no vol dir que no sigui útil com aplicació per preparar proves escrites. Trobem una oportunitat de mercat, ja que el **75% del conjunt no utilitza cap eina per estudiar proves de tipus test**, probablement per les conclusions de l'estudi de mercat previ en aquesta memòria. Finalment, podem destacar dos aspectes d'aquest perfil: el 75% d'ells prefereix obtenir el *feedback* de les respostes al contestar totes les preguntes, i també el 75% prefereix visualitzar totes les preguntes a la vegada. D'aquí podem deduir que la interfície, en la seva fase inicial, **ha de proporcionar el feedback al final de la prova, i ha de visualitzar totes les preguntes**.

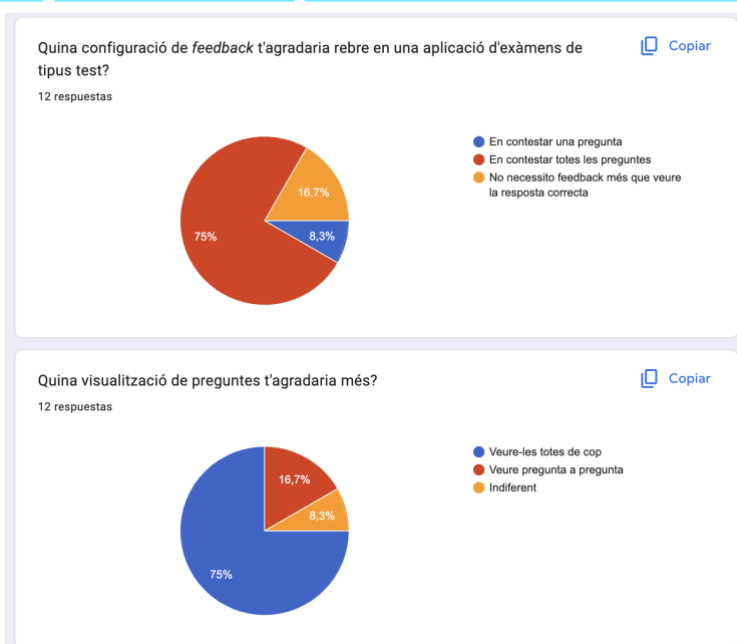


Fig. 6. Respostes rellevants perfil estudiant

Per acabar amb les enquestes, a la secció final podem destacar dues respostes que ens poden ajudar a prendre decisions sobre la monetització de l'aplicació de cara al futur (ja que inicialment volem oferir una solució 100% gratuïta), així com saber amb quina finalitat s'utilitzaria potencialment. Trobem que **només el 48,4% està disposat a pagar per utilitzar l'aplicació** i, d'aquests, el **46,7% li agradaria poder pagar de forma voluntària amb donació**. En relació amb la finalitat, el 35,5% la utilitzaria per preparar exàmens acadèmics, seguit de prop per preparar certificacions professionals (29%) i preparar oposicions (25,8%).

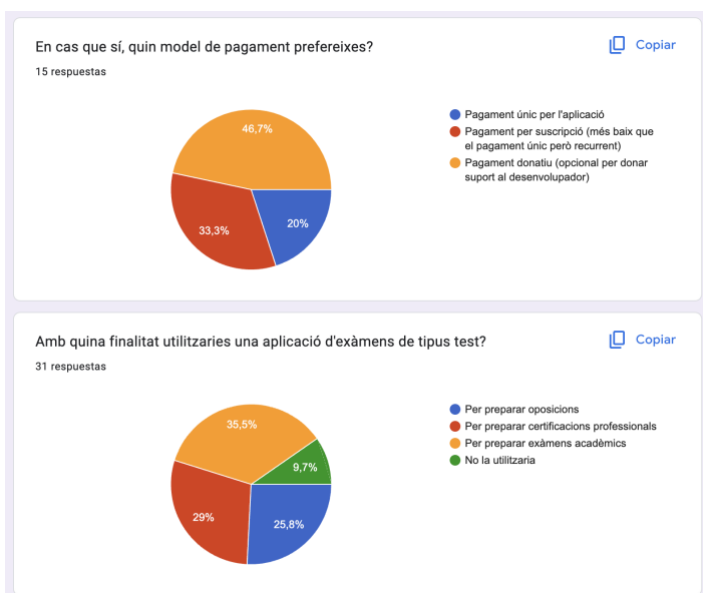


Fig. 7. Respostes generals de la part final

A l'apartat d'annexos es pot trobar el conjunt de les respostes individuals per analitzar-les amb tot detall, així com el conjunt de totes les estadístiques de les respostes.

2.2. Disseny conceptual

2.2.1. Arbre de navegació

A continuació visualitzarem un diagrama amb l'arbre de navegació que tindrà l'aplicació. Com que és una aplicació senzilla no necessitem diferenciar cap rol d'usuari, ens basta únicament amb l'inici de sessió per identificar-lo inequívocament i poder carregar els seus tests.

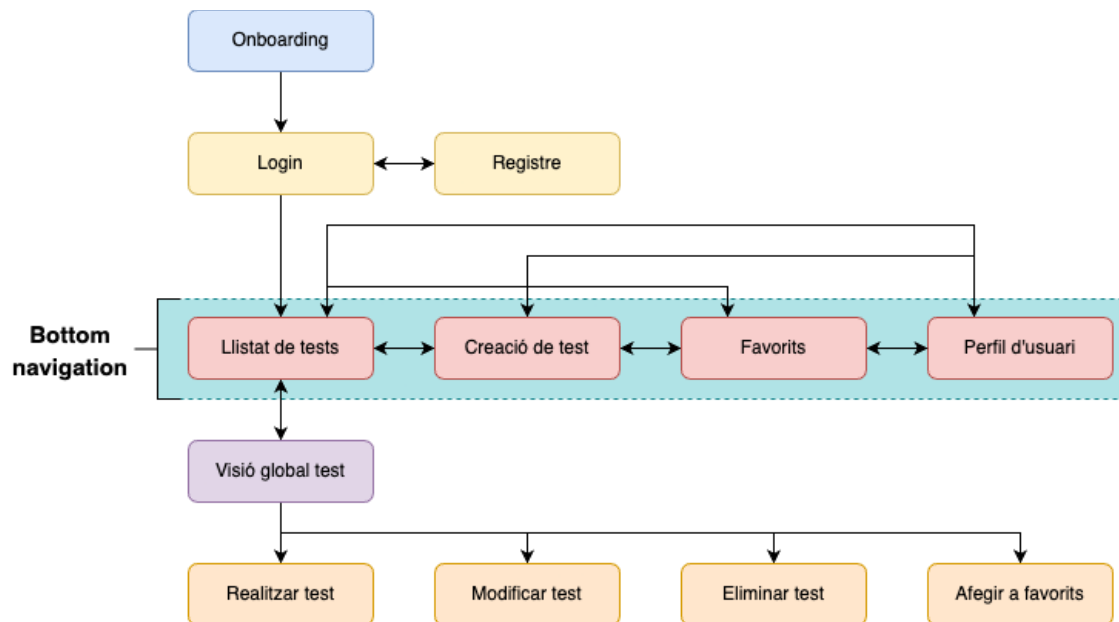


Fig. 8. Arbre de navegació de l'aplicació

Anem a fer una descripció breu de cada vista de l'arbre:

- **Onboarding:** quan l'usuari entri per primera vegada a l'aplicació, es mostrarà una introducció en forma de panells amb informació del que ofereix l'aplicació.
- **Login:** permetrà a l'usuari iniciar sessió si ja té compte, o bé navegar cap a la pantalla de registre per poder crear-ne una.
- **Registre:** permetrà a l'usuari enregistrar-se a l'aplicació, o bé tornar a la pantalla de login.
- **Bottom navigation:** la navegació principal de l'aplicació es farà utilitzar el disseny de *bottom* navigation, ja que les opcions que mostrarà estaran sempre disponibles des de qualsevol part de l'aplicació, i de manera ergonòmica [12]. Trobem les següents opcions:
 - **Llistat de tests:** un cop iniciada la sessió, l'usuari veurà el llistat de tests que ha creat, o bé una llista buida si no en té cap. És la pantalla d'inici de l'aplicació. En tocar/clicar damunt un test, el durà a la pantalla de visió global d'aquest.
 - **Creació de test:** permetrà a l'usuari definir un nou test.
 - **Favorits:** mostrarà els tests favorits de l'usuari, o bé una llista buida si no en té cap.
 - **Perfil d'usuari:** mostrarà la informació de l'usuari, així com li permetrà sortir de la sessió.

- **Visió global test:** mostrarà les dades bàsiques del test, juntament amb les accions que es podran realitzar:
 - **Realitzar test:** visualitza les preguntes del test i permet a l'usuari seleccionar les respostes.
 - **Modificar test:** permet modificar les preguntes i/o respostes del test.
 - **Eliminar test:** permet eliminar el test.
 - **Afegir a favorits:** permet afegir a favorits el test.

2.3. Prototipat

A partir dels esbossos d'interacció desenvolupats en la secció anterior, es prepararà un prototip horitzontal de l'aplicació que reflectirà les decisions de disseny preses d'acord amb les conclusions obtingudes en les etapes anteriors. Tot i que l'aplicació ha de ser multiplataforma, el disseny seguirà la filosofia de *mobile first*, per tant, el prototipat el farem agafant de base un dispositiu mòbil. Inicialment, es presentaran els esbossos en baixa fidelitat (*wireframes*) i s'aniran refinant fins a arribar als prototips finals en alta fidelitat (*mockups*). En la darrera secció, també es proporcionarà un resum de l'estil gràfic de l'aplicació, que inclourà el logotip, els colors i la tipografia utilitzats.

2.3.1. Prototipat Lo-Fi (*wireframes*)

En aquest apartat trobareu els *wireframes* de les pantalles principals del projecte. També, podeu consultar el projecte complet en l'aplicació Figma clicant [en aquest enllaç](#).

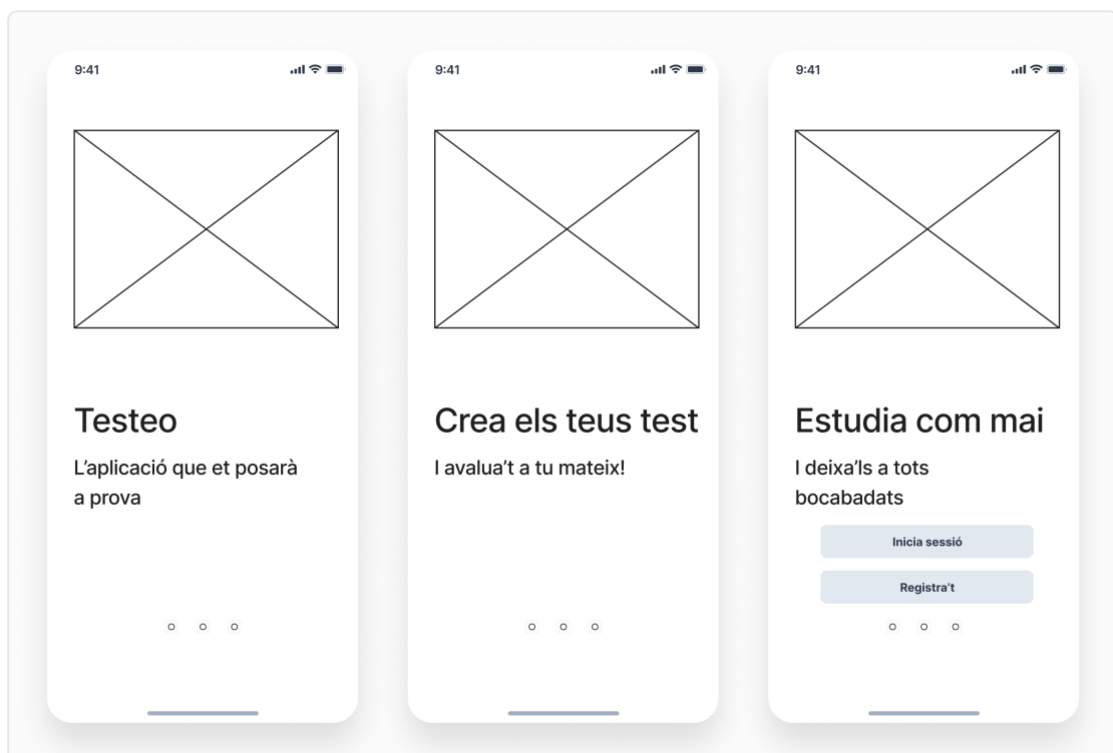


Fig. 9. Onboarding (*wireframes*)

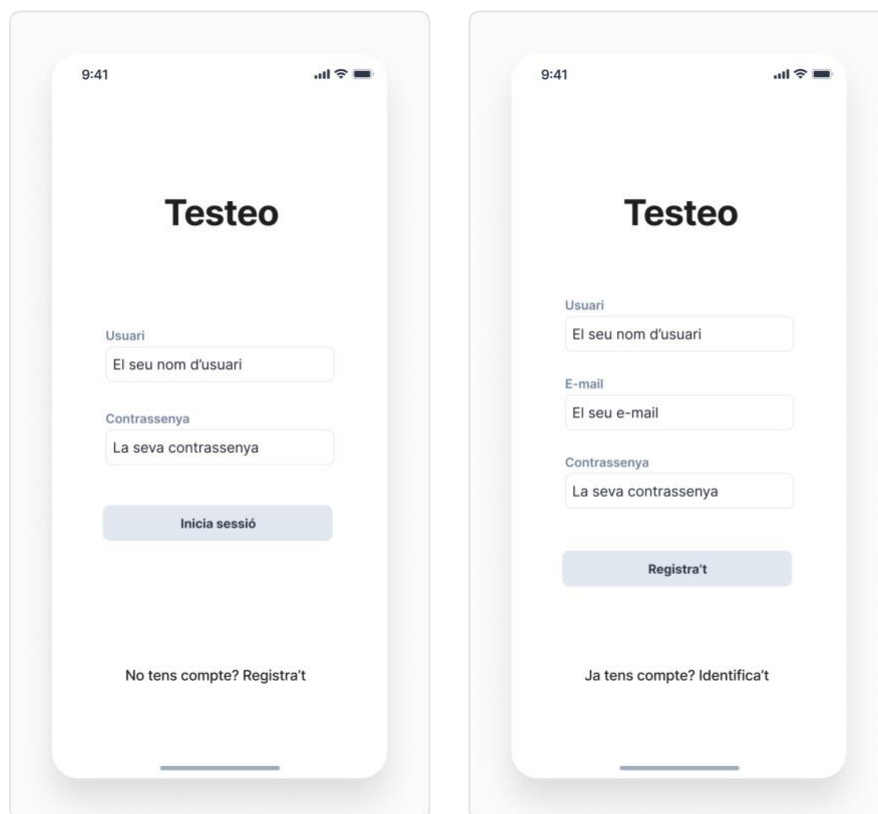


Fig. 10. Inici de sessió d'usuari i registre d'usuari (*wireframes*)

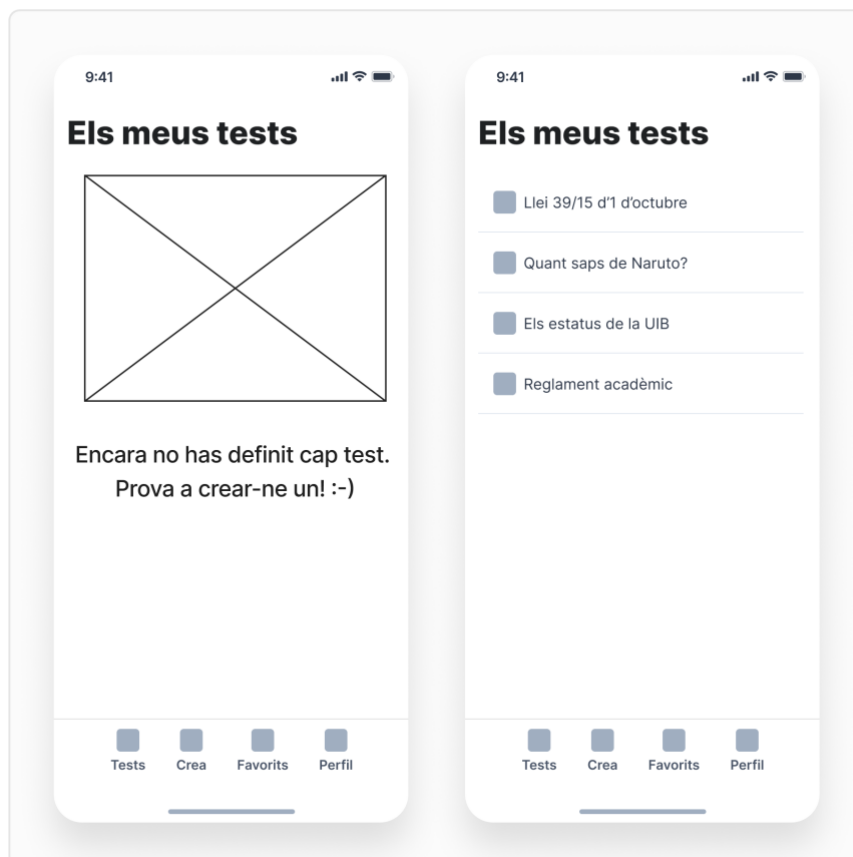


Fig. 11. Llista de test. Buida i amb dades (*wireframes*)

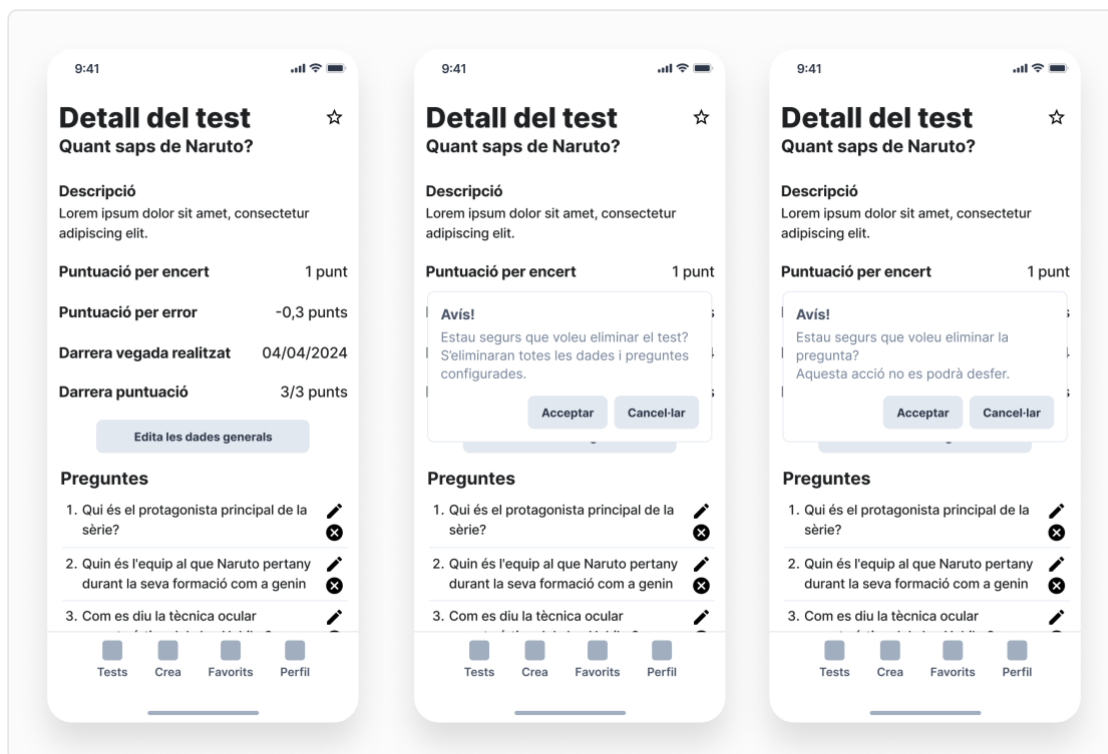


Fig. 12. Detall d'un test. Dades generals, confirmació d'eliminar test i confirmació d'eliminar pregunta (*wireframes*)

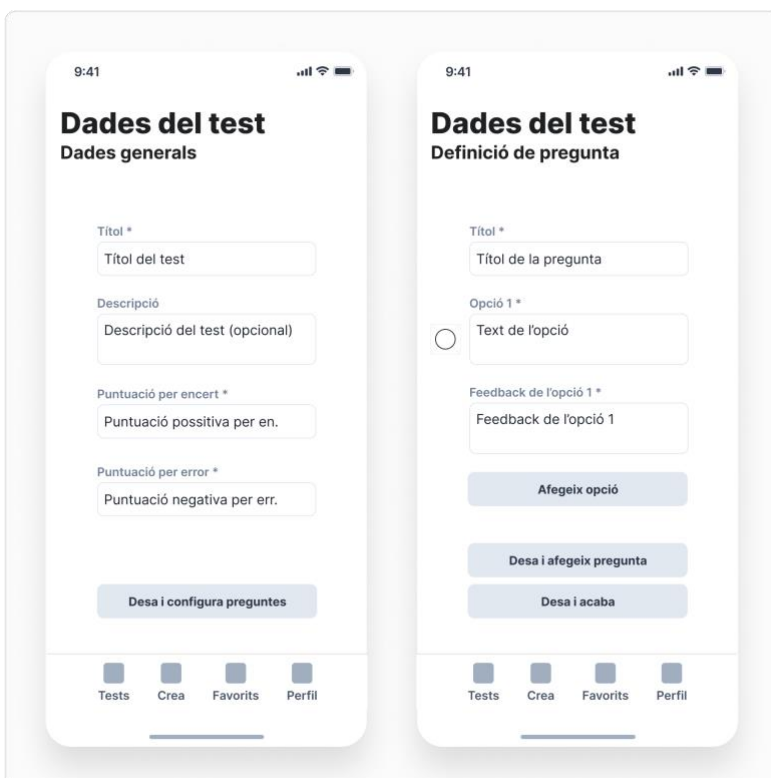


Fig. 13. Creació/modificació de test. Dades generals i de la pregunta (*wireframes*)

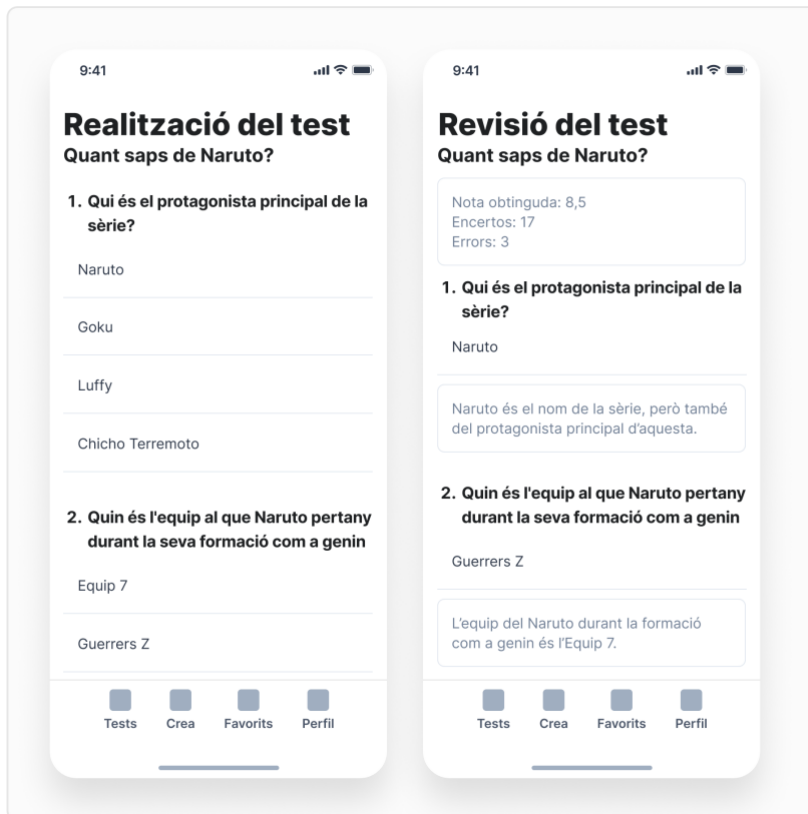


Fig. 14. Realització i revisió del test (wireframes)

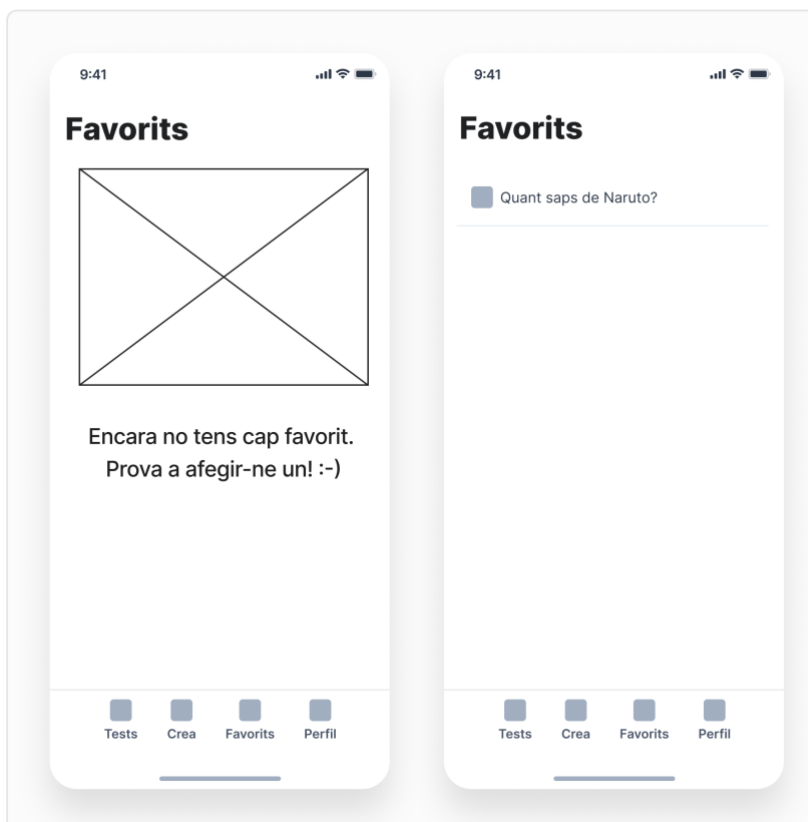


Fig. 15. Favorits. Buits i amb dades (wireframes)

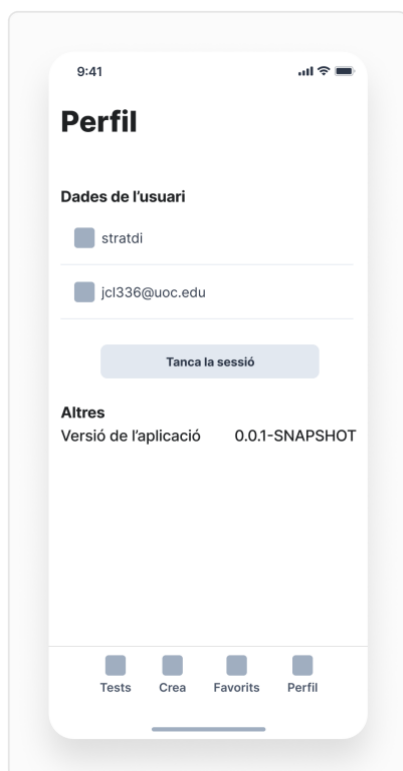


Fig. 16. Perfil d'usuari (*wireframes*)

2.3.2. Prototipat Hi-Fi (*mockups*)

Un com definits els *wireframes*, aquests s'han refinat per obtenir un prototip d'alta definició de les pantalles de l'aplicació. Els trobareu a continuació. També, podeu consultar el projecte complet en l'aplicació Figma clicant [en aquest enllaç](#).

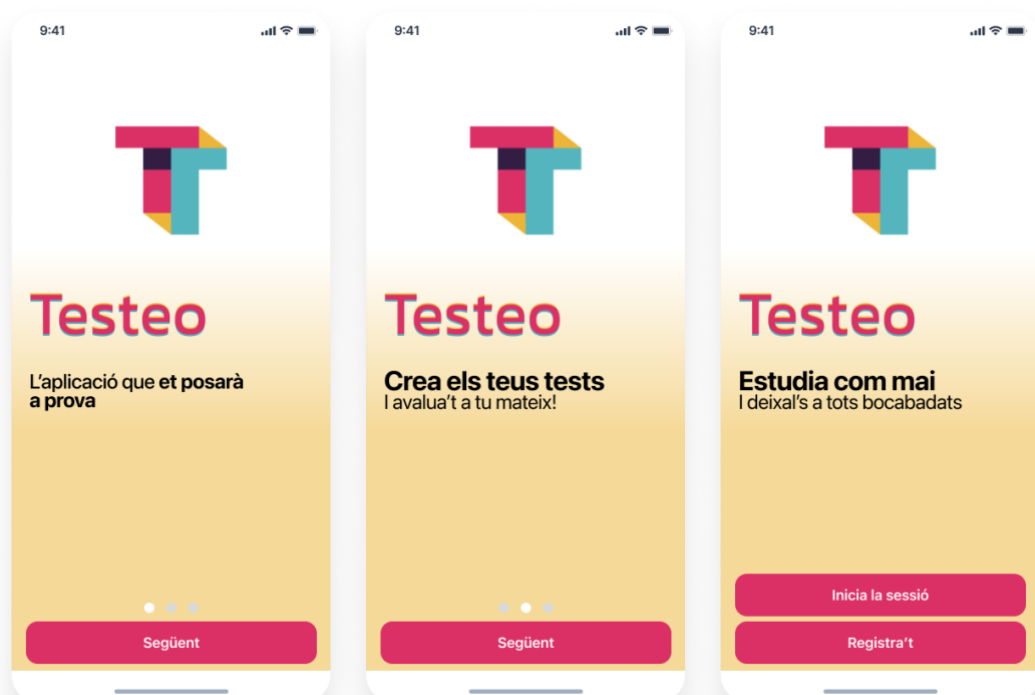


Fig. 17. Onboarding (*mockups*)

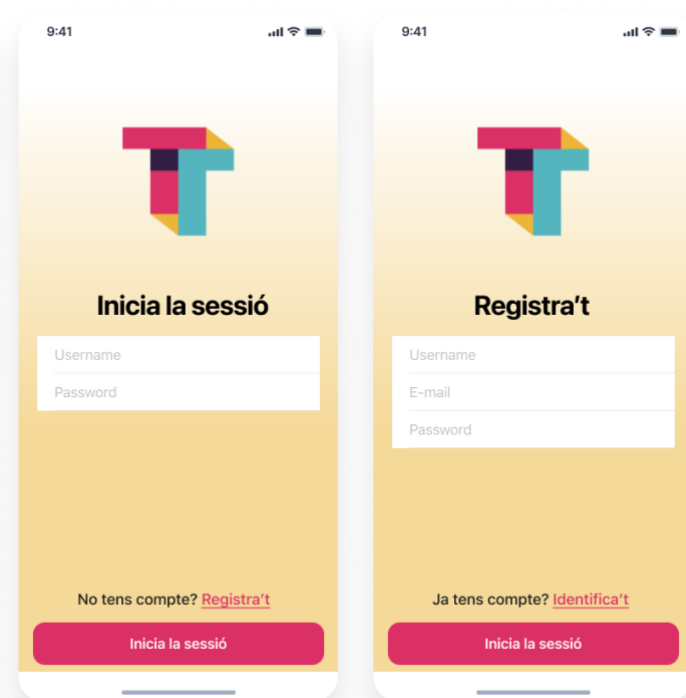


Fig. 18. Inici de sessió d'usuari i registre d'usuari (*mockups*)

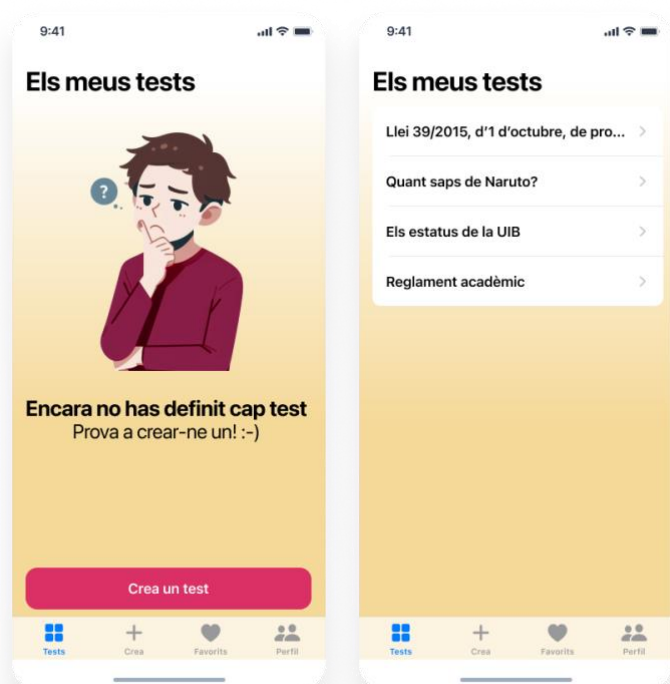


Fig. 19. Llista de test. Buida i amb dades (*wireframes*)

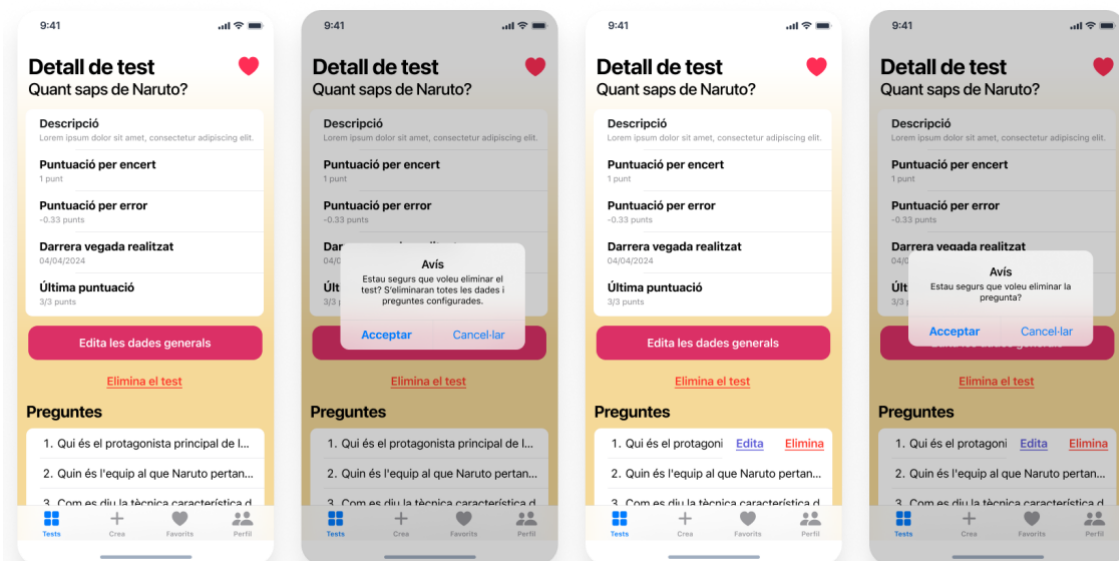


Fig. 20. Detall d'un test. Dades generals, confirmació d'eliminar test i confirmació d'eliminar test i pregunta (*mockups*)

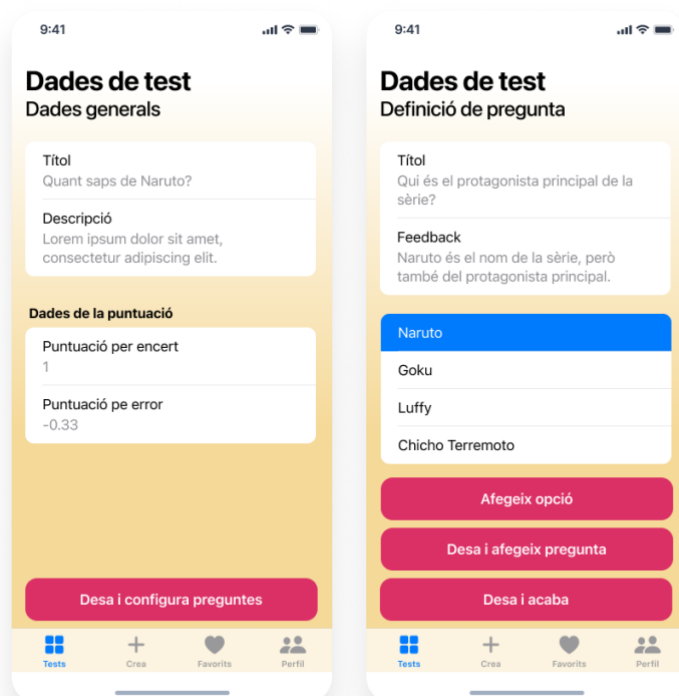


Fig. 21. Creació/modificació de test. Dades generals i de la pregunta (*mockups*)

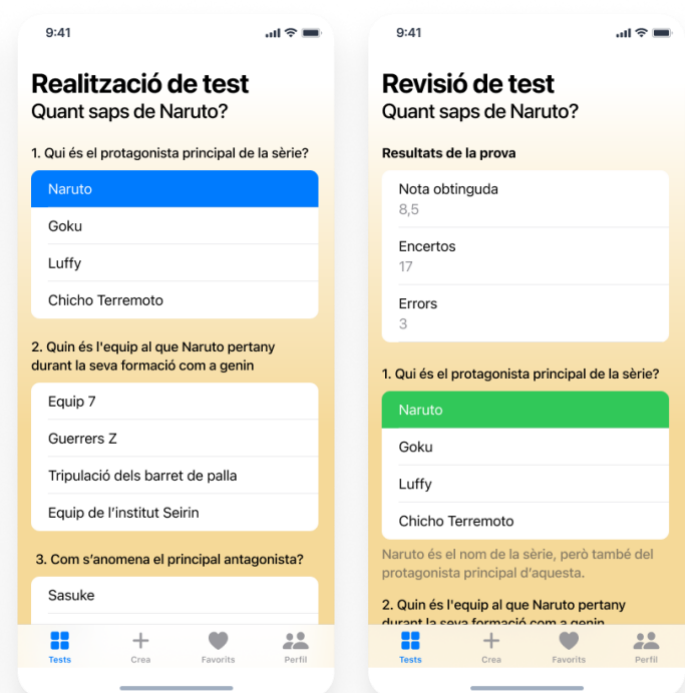


Fig. 22. Realització i revisió del test (*mockups*)

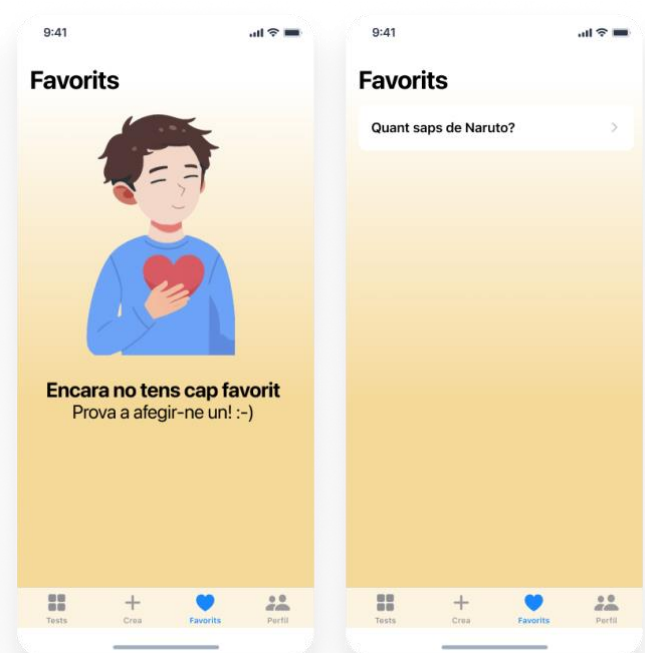


Fig. 23. Favorits. Buits i amb dades (*mockups*)

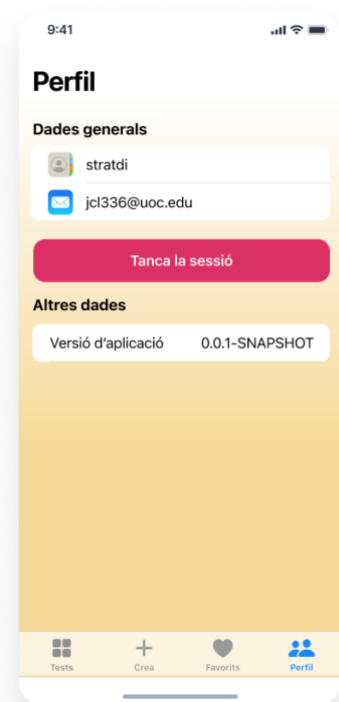


Fig. 24. Perfil d'usuari (*mockups*)

2.3.3. Decisions de disseny i recursos

2.3.3.1. Icona i logotip



Fig. 25. Icona de l'aplicació

S'ha dissenyat la icona per formar la lletra 't' inicial del nom de l'aplicació amb el que serien dos llapis formant un angle de noranta graus. El logotip utilitza els colors de la icona i forma la paraula 'Testeo'. S'ha utilitzat la font 'Kanit' disponible a [Google Fonts](https://www.google.com/fonts/specimen/Kanit).







Fig. 26. Logotip de l'aplicació

2.3.3.2. Paleta de colors

Els colors utilitzats, tant per la icona, com pel logotip, com dins la pròpia aplicació, són una derivació dels colors que ens proporciona el model de color CMYK, utilitzat en les empremtes i el món de la impressió. Aquesta decisió va sorgir d'agafar el model clàssic de fulla de respostes de la Universitat de les Illes Balears, on predomina la tonalitat magenta, per proporcionar a l'aplicació aquest estil de 'paper de respostes'. Cal dir que els colors no són pròpiament els de CMYK, sinó una derivació d'aquests cap a unes tonalitats més 'pastel' per fer-ho més agradable a la vista.

Els colors són els següents:

Taula VII - Colors de l'aplicació

Color	Nom	Codi Hexadecimal
	Magenta	#DB3568
	Cyan	#56B5BF
	Groc	#EDB539
	Negre	#FFFFFF

2.3.4. Avaluació

Per finalitzar el DCU s'ha d'avaluar el disseny dels prototips, per validar-lo o corregir-los si hi hagués mancances o deficiències importants, fent una nova iteració. Aquesta prova es recomana fer-la a un nombre igual o superior de cinc persones [13], però donat el temps reduït del que disposem únicament s'ha pogut dur a terme una iteració amb dos usuaris. Tot i així, s'ha obtingut un *feedback* important, que serà molt profitós de cara a la implementació de l'aplicació.

Per dur a terme l'avaluació, se'ls ha mostrat els prototips a l'aplicació de Figma i se'ls ha proposat fer les següents tasques:

- Crear un test.
- Llistar els tests.
- Marcar un test com a favorit.
- Realitzar un test.
- Eliminar un test.
- Tancar sessió.

Com podeu observar, s'ha obviat la part d'inici i registre d'usuari, ja que els usuaris es troben entrenats per utilitzar aplicacions a diari. També s'obvia l'edició d'un test, ja que són les mateixes pantalles que la creació. Abans de començar les tasques, expliquem a l'usuari la finalitat de l'aplicació de forma breu, amb l'objectiu que tingui una mica de context però que sigui ell qui esbrini el seu funcionament. Important també, que se li indica que la finalitat és avaluar el disseny, no la seva capacitat, i que el prototipat és limitat i sense interacció. Finalment, se li demana que digui en veu alta els pensaments, dubtes i altres que vagin sorgint durant la prova.

El *feedback* obtingut ha estat el següent:

- Troben poc intuïtiu marcar un test com a favorit amb un botó clàssic al detall del test. Es modifiquen els prototips per afegir el típic cor de *like*.
- Volen saber la versió de l'aplicació per saber si hi ha novetats. Aquesta retroalimentació no ens la esperàvem, ja que la persona que ho demana no és tècnica. Es modifiquen els prototips per afegir la versió al perfil d'usuari.
- Costa saber com tornar enrere per la navegació de l'aplicació. No s'inclou als *mockups* però es tindrà en compte en el disseny final.
- Es demana si es pot configurar test de resposta múltiple o de preguntes curtes. No és objectiu inicial de l'aplicació per aquest TFM, però és una bona millora de cara al futur.

3. Disseny tècnic

3.1. Arquitectura del sistema

L'aplicació desenvolupada utilitza la arquitectura client-servidor, essent el client el *frontend* i el servidor el *backend*. El *frontend* es desenvoluparà utilitzant el llenguatge de programació Angular, juntament amb el *Framework* Ionic, per aconseguir una visualització nativa segons el dispositiu que hi accedeixi. El *backend* es desenvoluparà utilitzant el llenguatge de programació Java, juntament amb Spring.

En el *backend* trobarem les següents tecnologies:

- MariaDB com a base de dades on emmagatzemar les dades.
- Hibernate com a ORM per facilitar el mapeig de la base de dades amb Spring.
- Apache Tomcat com a servidor HTTP.



Fig. 27. Tecnologies utilitzades

Tant la base de dades, el *backend* i el *frontend* estaran hostejats en un servidor nostre del qual disposem, ja que tenim total control sobre la màquina.

L'autenticació la farem mitjançant JWT, o JSON Web Tokens, que es generaran per part del *backend* davant una petició de login satisfactòria. Per emmagatzemar el token d'autenticació al *frontend* utilitzarem el plugin Capacitor Storage de Ionic, ja que aquesta mantindrà el token de manera persistent i no serà volàtil tal i com pot passar si emprem LocalStorage [14].

3.2. Casos d'ús

Els casos d'ús són una descripció detallada de la interacció dels usuaris amb l'aplicació per aconseguir un objectiu específic, de manera que serveix per documentar els requeriments. Aquests es centren en les accions de l'usuari i com ha de respondre el sistema, però només la part que l'usuari observarà. S'exclou la funcionalitat interna transparent per l'usuari.

Pel cas de l'aplicació només hi ha dos tipus de perfil: usuari sense autenticar i usuari autenticat. Per aquests darrers només hi ha un tipus de rol d'usuari, per tant, totes les tasques dels casos d'ús serà per aquest perfil.

A continuació trobareu el diagrama de casos d'ús que mostra la interacció que poden tenir els usuaris amb el sistema:

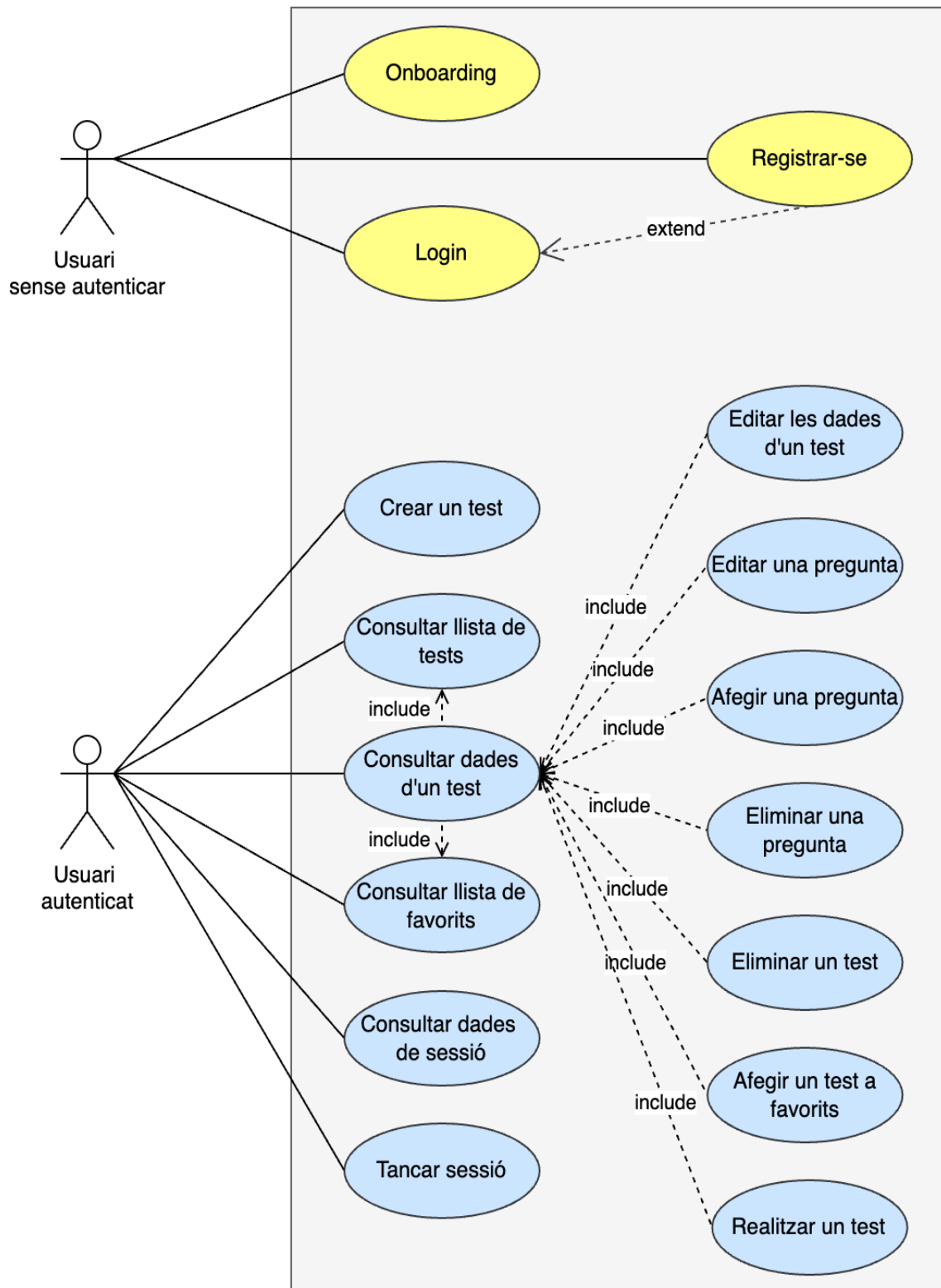


Fig. 28. Diagrama UML dels casos d'ús

Un cop presentat el diagrama UML, exposarem els diferents casos d'ús, juntament amb els actors que intervenen, les seves precondicions, els escenaris principals i alternatius d'èxit i les postcondicions. Cada cas d'ús anirà codificat per una ràpida identificació de cadascun.

Taula VIII - Onboarding (cas d'ús)

CU-01. Onboarding
Resum de la funcionalitat
Visualitzar la informació de l'aplicació i finalitzar-lo.
Actor
Usuari sense autenticar
Precondicions
L'usuari obre l'aplicació per primera vegada en el navegador.
Escenari principal
<ol style="list-style-type: none"> 1. L'usuari obre l'aplicació. 2. L'usuari fa el gest de <i>swipe</i>, o bé clica el botó de següent, fins arribar a la pantalla d'inici de sessió o registre. 3. Es desfa un registre per no tornar a mostrar el procés.
Postcondicions
No es tornarà a mostrar l' <i>onboarding</i> en aquest navegador.

Taula IX - Registrar-se (cas d'ús)

CU-02. Registrar-se
Resum de la funcionalitat
Permet que, donat un usuari sense cap compte d'usuari, es pugui crear-ne una.
Actor
Usuari sense autenticar.
Precondicions
No pot existir un usuari amb el mateix nom d'usuari ni el mateix correu.
Escenari principal
<ol style="list-style-type: none"> 1. El sistema mostra per pantalla el formulari de registre. 2. L'usuari introdueix el nom d'usuari, correu i contrasenya. 3. El sistema valida les dades i registra l'usuari.
Escenaris alternatius
<p>3.a. Ha aparegut un problema en el registre degut a les dades. Causes del problema:</p> <ul style="list-style-type: none"> • El nom d'usuari ja existeix. • El nom d'usuari no satisfà el mínim/màxim de caràcters. • El correu ja existeix • La contrasenya no satisfà el mínim/màxim de caràcters.
Postcondicions
El sistema mostra un avís de registre correcte i mostra la pantalla de login.

Taula X - Login (cas d'ús)

CU-03. Login
Resum de la funcionalitat
Permet a l'usuari identificar-se per accedir al sistema.
Actor
Usuari sense autenticar.
Precondicions
L'usuari ha de tenir un compte en el sistema.
Escenari principal
<ol style="list-style-type: none"> 1. El sistema mostra per pantalla el formulari de registre. 2. L'usuari introdueix el seu nom d'usuari i contrasenya. 3. El sistema valida les dades i identifica a l'usuari.
Escenaris alternatius
<ol style="list-style-type: none"> 3.a. L'usuari ha introduït dades incorrectes. <ol style="list-style-type: none"> 3.a.1. El sistema mostra un avís de dades incorrectes. 3.a.2. Tornar al punt 2 de l'escenari principal.
Postcondicions
El sistema identifica a l'usuari i el redirigeix a la pantalla principal.

Taula XI - Consultar llista de tests (cas d'ús)

CU-04. Consultar llista de tests
Resum de la funcionalitat
Mostra a l'usuari tots els tests que ha creat prèviament.
Actor
Usuari autenticat.
Precondicions
Cap.
Escenari principal
<ol style="list-style-type: none"> a. L'usuari inicia sessió en el sistema i és redirigit al llistat de tests. b. L'usuari clica a la icona del menú de Tests.
Escenaris alternatius
<ol style="list-style-type: none"> c. L'usuari no té cap test creat, se li mostrarà un llistat buit.
Postcondicions
L'usuari visualitza els tests que ha creat prèviament.

Taula XII - Crear un test (cas d'ús)

CU-05. Crear un test
Resum de la funcionalitat
L'usuari crea un nou test en el sistema.
Actor
Usuari autenticat.

Precondicions
Cap.
Escenari principal
<ol style="list-style-type: none"> 1. L'usuari clica la icona del menú de 'Crea'. 2. El sistema mostra el formulari de dades generals del test. 3. L'usuari completa les dades generals i clica al botó per afegir una pregunta. 4. El sistema mostra el formulari de creació de pregunta. 5. L'usuari completa les dades de la pregunta i: <ol style="list-style-type: none"> 5.a. L'usuari clica al botó de desar i acabar. 5.b. L'usuari clica al botó de desar i afegir pregunta, tornant al punt 4.
Escenaris alternatius
Cap.
Postcondicions
Es crea el test en el sistema i es posa disponible per l'usuari.

Taula XIII - Consultar dades d'un test (cas d'ús)

CU-06. Consultar dades d'un test
Resum de la funcionalitat
L'usuari vol conèixer els detalls del test o gestionar-lo.
Actor
Usuari autènticat.
Precondicions
<p>Ha de satisfer-se un d'aquests casos d'ús:</p> <ol style="list-style-type: none"> a. CU-04. Consultar llista de tests. b. CU-14. Consultar llista de favorits. <p>Ha d'existir un test per poder accedir-hi.</p>
Escenari principal
<ol style="list-style-type: none"> 1. L'usuari selecciona un test del llistat. 2. El sistema mostra les dades generals i les preguntes del test.
Escenaris alternatius
Cap.
Postcondicions
L'usuari veu les dades del test i pot gestionar-lo o realitzar-lo.

Taula XIV - Editar les dades d'un test (cas d'ús)

CU-07. Editar les dades d'un test
Resum de la funcionalitat
L'usuari vol editar les dades generals del test.
Actor
Usuari autènticat.
Precondicions
CU-06. Consultar dades d'un test.
Escenari principal

1. L'usuari clica damunt el botó 'Edita les dades generals'.
2. El sistema mostra un formulari amb les dades generals del test.
3. L'usuari modifica el contingut que escau.
4. L'usuari clica damunt el botó de 'Desa'.
5. El sistema actualitza les dades generals del test.

Escenaris alternatius

L'usuari surt del formulari, fent que no es realitzi cap canvi.

Postcondicions

L'usuari ha editat les dades generals del test.

Taula XV - Editar una pregunta (cas d'ús)

CU-08. Editar una pregunta

Resum de la funcionalitat

L'usuari vol editar les dades d'una pregunta del test.

Actor

Usuari autènticat.

Precondicions

CU-06. Consultar dades d'un test.

Escenari principal

1. L'usuari llisca la pregunta a l'esquerra i apareixen les opcions disponibles.
2. L'usuari clica damunt el botó 'Edita'.
3. El sistema mostra un formulari amb les dades de la pregunta del test.
4. L'usuari modifica el contingut que escau.
5. L'usuari clica damunt el botó de 'Desa'.
6. El sistema actualitza les dades de la pregunta del test.

Escenaris alternatius

L'usuari surt del formulari, fent que no es realitzi cap canvi.

Postcondicions

L'usuari ha editat les dades de la pregunta del test.

Taula XVI - Afegir una pregunta (cas d'ús)

CU-09. Afegir una pregunta

Resum de la funcionalitat

L'usuari vol afegir una pregunta al test.

Actor

Usuari autènticat.

Precondicions

CU-06. Consultar dades d'un test.

Escenari principal

1. L'usuari clica damunt el botó 'Afegeix pregunta'.
2. El sistema mostra un formulari amb les dades de la pregunta del test.
3. L'usuari indica la pregunta, el feedback i les respostes.
4. L'usuari marca la resposta que servirà per avaluar com a correcta.

5. L'usuari clica damunt el botó de 'Desa'.
6. El sistema afegeix les dades de la pregunta al test.

Escenaris alternatius

L'usuari surt del formulari, fent que no es realitzi cap canvi.

Postcondicions

L'usuari ha afegit una pregunta al test.

Taula XVII - Eliminar una pregunta (cas d'ús)

CU-10. Eliminar una pregunta

Resum de la funcionalitat

L'usuari vol eliminar una pregunta al test.

Actor

Usuari autènticat.

Precondicions

CU-06. Consultar dades d'un test.

Escenari principal

1. L'usuari llisca la pregunta a l'esquerra i apareixen les opcions disponibles.
2. L'usuari clica damunt el botó 'Elimina'.
3. El sistema mostra un avís de que l'acció no es podrà desfer.
4. L'usuari clica damunt l'opció 'Acceptar'.
5. El sistema elimina les dades de la pregunta del test.

Escenaris alternatius

L'usuari clica l'opció 'Cancel·lar' de l'avís, fent que no es realitzi cap canvi.

Postcondicions

L'usuari ha eliminat la pregunta del test.

Taula XVIII - Eliminar un test (cas d'ús)

CU-11. Eliminar un test

Resum de la funcionalitat

L'usuari vol eliminar un test.

Actor

Usuari autènticat.

Precondicions

CU-06. Consultar dades d'un test.

Escenari principal

1. L'usuari clica damunt el botó 'Afegeix pregunta'.
2. El sistema mostra un avís de que l'acció no es podrà desfer.
3. L'usuari clica damunt l'opció 'Acceptar'.
4. El sistema elimina el test i redirigeix a l'usuari a la llista de tests.

Escenaris alternatius

L'usuari clica l'opció 'Cancel·lar' de l'avís, fent que no es realitzi cap canvi.

Postcondicions

L'usuari ha eliminat el test.

Taula XIX - Afegir un test a favorits (cas d'ús)

CU-12. Afegir un test a favorits

Resum de la funcionalitat

L'usuari vol afegir un test als seus favorits.

Actor

Usuari autènticat.

Precondicions

CU-06. Consultar dades d'un test.

Escenari principal

1. L'usuari clica damunt la icona amb forma de cor.
2. El sistema marca el test com a favorit.

Escenaris alternatius

Cap.

Postcondicions

L'usuari ha afegit un test als seus favorits.

Taula XX - Realitzar un test (cas d'ús)

CU-13. Realitzar un test

Resum de la funcionalitat

L'usuari vol realitzar un test per posar a prova els coneixements.

Actor

Usuari autènticat.

Precondicions

CU-06. Consultar dades d'un test.

Escenari principal

1. L'usuari clica damunt el botó de 'Realitza el test'.
2. El sistema mostra el formulari totes les preguntes juntament amb les seves respostes.
3. L'usuari marca les respostes que creu que són correctes.
4. L'usuari clica el botó de 'Finalitza el test'.
5. El sistema mostra la pantalla de resultat indicant la nota obtinguda, així com el feedback per cada pregunta.

Escenaris alternatius

L'usuari surt del formulari, fent que no es realitzi cap canvi.

Postcondicions

L'usuari ha realitzat el test i ha rebut feedback.

Taula XXI - Consultar llista de favorits (cas d'ús)

CU-14. Consultar llista de favorits
Resum de la funcionalitat
Mostra a l'usuari tots els tests que ha afegit prèviament com a favorits.
Actor
Usuari autènticat.
Precondicions
Cap.
Escenari principal
L'usuari clica a la icona del menú de 'Favorits'.
Escenaris alternatius
L'usuari no té cap test marcat com a favorit, se li mostrarà un llistat buit.
Postcondicions
L'usuari visualitza els tests que ha marcat prèviament com a favorits.

Taula XXII - Consultar dades de sessió (cas d'ús)

CU-15. Consultar dades de sessió
Resum de la funcionalitat
L'usuari vol consultar les seves dades de la sessió de l'aplicació.
Actor
Usuari autènticat.
Precondicions
Cap.
Escenari principal
L'usuari clica a la icona del menú de 'Perfil'.
Escenaris alternatius
Cap.
Postcondicions
L'usuari visualitza les seves dades de la sessió.

Taula XXIII - Tancar sessió (cas d'ús)

CU-16. Tancar sessió
Resum de la funcionalitat
L'usuari tanca la sessió iniciada a l'aplicació.
Actor
Usuari autènticat.
Precondicions
CU-15. Consultar dades de sessió.
Escenari principal
CU-15. Consultar dades de sessió.

Escenaris alternatius

1. L'usuari clica damunt el botó de 'Tanca la sessió'.
2. El sistema esborra les dades de sessió.
3. El sistema redirigeix a l'usuari a la pantalla d'inici de sessió.

Postcondicions

L'usuari ha tancat la sessió.

3.3. Base de dades

Independentment de la base de dades relacional que utilitzem, que com hem esmentat prèviament seria una base de dades MariaDB, començarem a definir-la realitzant el seu diagrama d'entitat-relació.

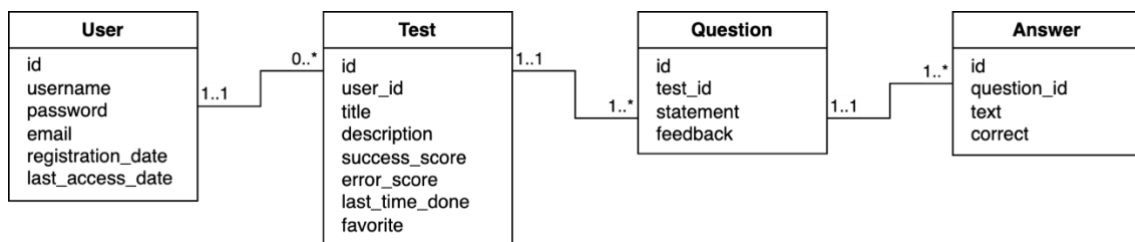


Fig. 29. Diagrama Entitat-Relació de l'aplicació

La taula 'User' serveix per enregistrar els usuaris a l'aplicació i, a més, per relacionar tots els tests que generin. La taula 'Test' representarà les dades generals d'aquest, i estarà relacionada amb la taula 'Question', on hi figurarà les dades de la pregunta. Cada 'Question' estarà relacionada amb un únic test. Finalment, la taula 'Answer' tindrà les dades d'una opció d'una pregunta en concret.

4. Disseny tècnic

4.1. Decisions de disseny

Per la part de *backend* s'ha decidit utilitzar la darrera versió d'Spring Boot que hi havia en el moment de començar-lo, la versió 3.2.4. Aquesta versió introdueix canvis rellevants, sobretot a la part de seguretat, ja que ara tot es basa en filtres. Per això, hem hagut d'actualitzar els coneixements sobre aquesta part, ja que els coneixements previs eren de la versió 2.6.1, on la seguretat es fa amb l'extensió de classes. També, s'ha decidit utilitzar la darrera versió LTS de Java, és a dir, la versió 21.

En aquest punt hem hagut de decidir com fer l'autenticació, ja que és el primer projecte que fem on el *backend* i el *frontend* es comuniquen via API, ja que totes les aplicacions fetes fins ara eren fetes amb *Thymeleaf*, que ens permet acoblar la presentació amb el codi Spring, i s'integra perfectament amb l'autenticació d'usuaris amb Spring Security. Un cop investigat com fer-ho, es decideix fer l'autenticació mitjançant JWT. La teoria és simple: el *frontend* cridarà a un *endpoint* de l'API amb l'usuari i password i el *backend*, si l'autenticació és correcta, generarà un *token* amb una caducitat d'un mes. Aquest *token* es retornarà per l'API però també s'afegirà a una cookie, per tal de facilitar l'ús de l'API via navegador web. Un cop rebut el *token*, el *frontend* el desa en memòria mitjançant *Capacitor Storage* i el fa servir com a capçalera a les peticions a l'API per autenticar a l'usuari.

Pel *frontend*, s'ha decidit emprar la darrera versió disponible d'Ionic , que en el moment de començar el desenvolupament era la versió 7.5.0 per Ionic-Angular.

Finalment, la instal·lació de tot l'*stack* es farà en un servidor de la nostra propietat, i per facilitar la posada en marxa s'inicialitzarà amb Docker, concretament amb un fitxer *compose*. S'enregistrarà un subdomini del domini nostre actual, cruzl.dev.

4.2. Llibreries utilitzades

Pel desenvolupament de l'aplicació s'han utilitzat amb diferents llibreries pel *backend* i el *frontend*. Anem a llistar-les i a motivar el seu ús.

Backend

- **mysql-connector-j**: controlador JDBC per fer la connexió amb bases de dades MySQL/MariaDB.
- **spring-boot-starter-jpa**: utilitzat per integrar Spring Data JPA pels repositoris i operacions de base de dades amb JPA.
- **spring-boot-starter-web**: utilitzat per desenvolupar els *endpoints*.
- **spring-boot-starter-security**: utilitzat per l'autenticació d'usuaris al *backend*.
- **lombok**: utilitzat per generar getters, setters, constructors, etc.

- **jwt-api/jjwt-impl/jjwt-jackson**: utilitzats per la gestió completa dels JWT i la seva autenticació/validació.
- **springdoc-openapi-starter-webmvc-ui**: utilitzat per generar la documentació de l'API REST en format OpenAPI així com una web per poder utilitzar-los.
- **mapstruct**: utilitzar per la conversió entre classes, generalment entre *Entity* i *DTO*.

Frontend

- **ionic**: utilitzat pels components de presentació, així com per les rutes entre pestanyes.
- **ionicons**: utilitzat per les icones de l'aplicació.
- **capacitor**: utilitzat per desar de forma no volàtil el JWT.
- **swiper**: utilitzat per desplaçar-se entre pantalles en l'*onboarding*.

4.3. Desenvolupament

4.3.1. Inici

Primer de tot, hem creat un repositori a GitHub per tenir el control de versions del projecte. Seguidament, l'hem clonat a la nostra màquina per poder-hi fer feina. Un cop fet, anem al directori del repositori per començar a crear l'aplicació.

Per crear l'esquelet del projecte amb Ionic Angular, un cop hem instal·lat iònic-clí, hem d'executar la comanda següent:

```
ionic start testeo tabs --type=angular -capacitor
```

'testeo' és el nom de l'aplicació, i el paràmetre següent indica que inicialitzi els components per generar una presentació principal de tipus pestanyes. El paràmetre *type* és per indicar a ionic que utilitzarem Angular, i finalment el darrer paràmetre és per instal·lar Capacitor juntament amb el projecte.

A continuació, generarem les pàgines que tindrà l'aplicació amb les següents comandes:

```
ionic g page pages/intro
ionic g page pages/login
ionic g page pages/test-list
ionic g page pages/test-create
ionic g page pages/test-edit
ionic g page pages/test-detail
ionic g page pages/test-favorites
ionic g page pages/test-take
ionic g page pages/question-create
ionic g page pages/question-edit
ionic g page pages/user-profile
```

Movem el component 'tab' al directori de *pages* per tenir-ho tot al mateix punt. També necessitarem crear dos *services*: un per l'autenticació i un altre per fer les crides a l'API REST. Crearem un tercer per la gestió de missatges d'estat. Els crearem amb les següents comandes:

```
ionic g services services/authentication
ionic g services services/test
ionic g services services/toast
```

Finalment, crearem quatre components reutilitzables entre pantalles, per tal de satisfer el principi DRY. Aquests components tenen la particularitat que rebran la informació de diferents bandes però amb visualització i operativa comuna.

Els crearem amb les següents comandes:

```
ionic g component shared/components/detail
ionic g component shared/components/list
ionic g component shared/components/question-form
ionic g component shared/components/take-form
ionic g component shared/components/test-form
```

4.3.2. Llista de tests

Des d'aquesta pantalla es visualitzaran tots els tests de l'usuari. Si no n'hi ha cap, es visualitza un missatge amigable per avisar a l'usuari que el llistat és buit per aquest motiu. Fent clic damunt el nom del test podrem accedir al seu detall, o bé, si lliscam el nom cap a l'esquerra ens apareixerà un accés ràpid a l'opció d'eliminar el test. L'eliminació de test implicarà sempre un missatge de confirmació per realitzar l'acció, de manera que previndrà esborrats per error. Aquesta pantalla utilitzarà el component compartit **list**, ja que la visualització i accions són les mateixes que trobarem a la pantalla de favorits.

4.3.3. Detall de test

En aquesta pantalla es visualitzen totes les dades d'un test, així com les preguntes de les que disposa. També, permet editar les dades generals, eliminar el test, o bé afegir una nova pregunta a la bateria de preguntes. També ens permetrà afegir el test als favorits clicant la icona del cor ubicat a la dreta de la capçalera. Lliscant una pregunta a l'esquerra trobem dues accions per realitzar sobre una pregunta: editar-la i eliminar-la. Aquesta pantalla reutilitza el component **details**, ja que també es visualitzarà el mateix des del detall dels tests favorits.

4.3.4. Edició de test

En aquesta pantalla editarem les dades generals del test: text, descripció, i les diferents puntuacions per avaluar el test. A part d'això, ens permetrà generar una nova pregunta un cop haguem desat els canvis. Aquesta pantalla reutilitza el component **test-form**, ja que s'utilitzarà el mateix formulari per la creació i edició, així com per l'edició d'un formulari favorit.

4.3.5. Alta de test

En aquesta pantalla crearem un nou test especificant les dades generals del test: text, descripció, i les diferents puntuacions per avaluar el test. A part d'això, ens permetrà generar una nova pregunta un cop haguem desat els canvis. Aquesta pantalla reutilitza el component **test-form**, ja que s'utilitzarà el mateix formulari per la creació i edició, així com per l'edició d'un formulari favorit.

4.3.6. Alta de pregunta

En aquesta pantalla podrem definir una nova pregunta per un test existent. En aquesta definició, s'haurà d'especificar el text de la pregunta, el *feedback* per quan es faci la correcció, i les preguntes disponibles, així com quina d'elles és la correcta. Es decideix establir un mínim de dues preguntes, ja que no té cap sentit generar una pregunta de tipus tests amb menys de dues preguntes per contestar. Aquesta pantalla reutilitza el component **question-form**, ja que utilitzarà el mateix formulari per la creació i edició de preguntes.

4.3.7. Edició de pregunta

En aquesta pantalla editarem una pregunta modificant les seves dades, així com les diferents preguntes que la componen. Aquesta pantalla reutilitza el component **question-form**, ja que utilitzarà el mateix formulari per la creació i edició de preguntes.

4.3.8. Llista de favorits

Des d'aquesta pantalla es visualitzaran tots els tests marcats com a favorits per l'usuari. Si no n'hi ha cap, es visualitza un missatge amigable per avisar a l'usuari que el llistat és buit per aquest motiu. Fent clic damunt el nom del test podrem accedir al seu detall, o bé, si lliscam el nom cap a l'esquerra ens apareixerà un accés ràpid a l'opció d'eliminar el test. L'eliminació de test implicarà sempre un missatge de confirmació per realitzar l'acció, de manera que previndrà esborrats per error. Aquesta pantalla utilitzarà el component compartit **list**.

4.3.9. Realització de test

Des d'aquesta pantalla es visualitzaran totes les preguntes del test amb les seves corresponents respostes. Fent clic damunt la resposta que volem es marcarà per la seva avaluació. L'avaluació no es realitzarà fins que l'usuari finalitzi la realització amb el botó al final de la pantalla. Un cop finalitzada, la mateixa pantalla mostra el resultat de l'avaluació i mostra el *feedback* de les preguntes a l'usuari, així com la nota obtinguda. Aquesta pantalla utilitzarà el component compartit **take-form**.

4.3.10. Autenticació

S'ha hagut d'ajustar l'autenticació per part del *backend*, ja que Spring per defecte utilitza les seves llibreries d'Spring Security, i fa el maneig de sessions. A l'emprar JWT, tal com hem explicat anteriorment, calia casar la llibreria JWT amb Spring Security. La manera de procedir ha estat la següent:

- Es configura Spring Security per permetre qualsevol petició a l'endpoint de *login* o de *register*, ja que mai podríem enregistrar-nos o bé autenticar-nos.
- També es configura per fer una sessió *stateless* per garantir que no es crea una sessió, perquè totes les crides es faran via API.
- Un cop s'utilitza l'endpoint de *login*, es genera un token JWT i es retorna perquè el *frontend* en pugui fer ús. També es posa com a *cookie* per facilitar l'ús de l'utilitat de Swagger.
- El *frontend* recull aquest token i el desa amb *Capacitor Storage*, de manera que queda en la sessió de l'aplicació.
- Per cada crida que faci el *frontend* al , enviarà aquest *token* com a *header* de la petició. El *backend*, gràcies a un filtre d'Spring, interceptarà cada petició i comprovarà si hi figura el *token* al *header* de la petició, o bé com a *cookie*. En cas contrari, denegarà el recurs. També validarà la validesa del *token*, denegant el recurs si ha caducat.
- El *frontend* valida abans de cada petició la validesa del *token* i, si ha caducat, tanca sessió i redirigeix a la pantalla de *login*.

També s'ha implementat un mecanisme d'*autologin*. Això vol dir que, si l'usuari té una sessió iniciada i tanca el navegador web, quan torni a obrir-lo i entrar dins l'aplicació aquesta consultarà el *Capacitor Storage* per consultar si té en memòria un *token* i, d'aquesta manera, completar l'inici de sessió automàticament.

4.3.11. Onboarding

El procés d'*onboarding* només ens interessa que surti la primera vegada que s'entra dins l'aplicació. Per fer aquest control, també hem utilitzat *Capacitor Storage*. Quan s'entra dins l'aplicació, es consulta en memòria si la variable *INTRO_KEY* té el valor *true*. Si aquesta condició se satisfà, no mostrarà l'*onboarding*. En cas contrari, redirigeix a les pantalles de l'*onboarding*. Un cop finalitzat, actualitzarà la variable a *true* per fer que futures vegades no el mostri.

4.4. Punts de control

El desenvolupament de l'aplicació es du a terme en les PAC3 i PAC4. Per tal de dur un seguiment per cadascuna, es farà una revisió de l'aplicació per cada PAC.

4.4.1. Revisió de la PAC3

En aquest moment de la revisió, l'aplicació es troba bastant completa però inacabada. S'ha pogut dur a terme gran part del desenvolupament de les funcionalitats, tot i que encara queda l'objectiu principal d'aquesta aplicació: realitzar un test i ser avaluat. També, el disseny és el bàsic que proporciona Ionic inicialment, i cal encara definir els estils diferenciadors. També, falta realitzar la

gestió de llistes buides, ja que actualment apareixerà un component *skeleton* indefinidament.

Per tant, recapitulant, en aquesta revisió l'usuari pot fer:

- Registre d'usuari.
- Inici de sessió.
- Creació de test.
- Edició de test.
- Visualitzar el llistat de tests.
- Visualitzar els tests favorits.
- Crear una pregunta.
- Editar una pregunta.
- Visualitzar les dades d'usuari.
- Tancar sessió.

De cara al lliurament final, falta:

- Realitzar un test.
- Visualitzar el *feedback* de la realització del test.
- Fer l'*onboarding* de l'aplicació.
- Fer compose de Docker.

4.4.2. Revisió de la PAC4

Durant l'execució d'aquesta PAC hem detectat un *bug* a la pantalla de creació/edició de pregunta: si s'esborra una de les possibles respostes, la pregunta seleccionada com vertadera no manté l'índex correctament i es desplaça. Això és degut a un mal càlcul de l'índex de resposta esborrada. S'ha corregit per aquesta PAC.

També ens vam adonar d'un problema que no havíem previst: les pestanyes estan dissenyades per mantenir la vista actual, fins i tot si es canvia a una altra pestanya. Això ens porta a una qüestió important: què passa si obrim un test des del llistat, canviem a la pestanya de favorits, l'eliminem des d'allà, i tornar a la primera pestanya? Doncs continuava mostrant la informació del test un cop esborrat.

Per tal de garantir que aquestes situacions no es produïssin, es va modificar el comportament de càrrega de la informació. En lloc de realitzar la càrrega al constructor o al *ngOnInit*, es va traslladar al mètode *ionViewWillEnter*, disponible a qualsevol pàgina d'Ionic. Aquest mètode s'executa cada vegada que l'usuari entra a la pàgina. Així, si en tornar a consultar la informació del test aquest ja no està disponible, es farà una redirecció automàtica a la pàgina de la pestanya pertinent.

Altres desenvolupaments complementaris van ser generar dos *Dockerfile*: un per al i un per al *frontend*; i un *Docker compose* per tal d'engegar aquests dos darrers serveis més una base de dades. D'aquesta manera simplifiquem la posada en marxa i dependències i afavorim a la portabilitat de la infraestructura de l'aplicació. Cal destacar que, gràcies a *Hibernate*, podem crear tots els objectes necessaris a la base de dades de manera automàtica en engegar l'aplicació. Això ho aconseguim amb la propietat *spring.jpa.hibernate.dd-auto: update*.

Finalment, s'ha desenvolupat la pantalla de realització d'un test, així com la seva anàloga de visualitzar el *feedback* un cop enviat per avaluar. L'avaluació inicialment es pensava fer al *backend*, però finalment hem decidit fer-ho al *frontend* per alleugerar la càrrega del servidor. També s'ha desenvolupat l'*onboarding* de l'aplicació.

A part de la correcció de *bugs* i darreres implementacions, s'ha fet una feina de *refactoring* per optimitzar el codi i eliminar duplicitats i codi redundant.

4.5. Errors del programari

Dins la darrera PAC s'ha fet un refinament per detectar possibles *bugs*, però n'hi ha d'altres que han quedat pendents de resoldre, ja que no tenen un impacte significant o bloquejant per l'usuari final. Tot i això no vol dir que siguin menys importants, així que cal posar-hi remei com més aviat millor.

- **Registre d'usuari ja existent.** Si es prova d'enregistrar un usuari amb un nom d'usuari existent, o bé una adreça de correu ja enregistrada, mostra un error genèric. Convindria especialitzar l'error per informar a l'usuari d'on es troba l'error.
- **Temps de caducitat del *token*.** Si bé l'aplicació es troba configurada per generar *tokens* amb validesa d'un mes, de vegades ocorre que el detecta com a expirat i tanca la sessió. S'hauria de revisar per evitar que l'usuari s'hagi d'autenticar sense causa justificada.

4.6. Execució del codi

Per executar el codi no hi ha gaires requisits, només cal tenir instal·lat Docker. Si no ho tenim ja, podem descarregar-lo i seguir les passes d'instal·lació indicades en la [documentació oficial de Docker](#).

Un cop realitzat això, obrim una terminal i ens dirigim al directori on es troba el codi font. Finalment, executem amb la comanda següent:

```
docker compose up
```

Aquesta comanda construirà el *backend* i el *frontend* executant els seus fitxers *Dockerfile*, i es descarregarà una imatge amb una instància de *mariadb*. Finalment, podrem trobar disponibles les aplicacions en els següents enllaços:

- **Testeo (*frontend*):** <http://localhost:7100>
- **Testeo (*backend*):** <http://localhost:7000>
- **Swagger UI:** <http://localhost:7000/swagger-ui/index.html>

Per si no fos possible executar el codi per restriccions a la màquina, o bé per no tenir-hi Docker instal·lat, s'ha desplegat en un servidor local tot el necessari per poder accedir-hi des de qualsevol banda.

- **Testeo (*frontend*)**: <https://testeo.cruzl.dev>
- **Testeo (*backend*)**: <https://testeo-backend.cruzl.dev>
- **Swagger UI**: <https://testeo-backend.cruzl.dev/swagger-ui/index.html>

El codi també es pot obtenir des del repositori de GitHub on s'hosteja. El podeu trobar en el següent enllaç:

- **GitHub repo**: <https://github.com/stratdi/Testeo>

5. Conclusions i treball futur

Com a punt i final del projecte, volem presentar les conclusions obtingudes del desenvolupament d'aquest projecte, així com les possibles característiques futures que es podrien incorporar a l'aplicació desenvolupada.

5.1. Conclusions del treball

Al principi del desenvolupament em vaig sentir còmode desenvolupant el amb Spring, ja que és el meu camp i no vaig tenir dificultats. Sí que em vaig trobar que era la primera vegada que treballava separant el *frontend* del *backend* i, per tant, tenia la barrera de com gestionar l'autenticació, ja que Spring Security ho integra molt bé quan s'empra juntament amb Spring Web. Per això, investigant com es fa pel món, vaig topar amb els *token JWT*, i vaig decidir implementar-ho d'aquesta manera. No em va agradar gaire, per no dir gens, fer feina d'aquesta manera: Spring Security és una llibreria robusta i provada, i amb els *token JWT* s'ha de fer una sessió *stateless* i gestionar tu mateix l'autenticació.

Per altra banda, en haver de fer feina amb dues tecnologies vaig decidir abandonar Eclipse per emprar VSCode pel *backend* i el *frontend*, i he de dir que m'ha sorprès gratament veure l'agilitat, rapidesa i eines de depuració que inclou, fins i tot per Spring.

En relació amb la tecnologia del *frontend*, Angular m'ha sorprès també per la facilitat que es poden construir pantalles, tot i que també part del mèrit la té Ionic Framework. És el que més m'ha agradat del *frontend*: de primeres et dona una presència d'aplicació impressionant, sembla nativa. Juntament amb la bona documentació i les facilitats per desenvolupar, amb exemples, han fet que aquest desenvolupament sigui molt plaent. Després d'aquest treball continuaré aprenent a utilitzar-la, ja que ofereix moltes integracions i components.

Pel que fa als objectius plantejats es donen per satisfets tots, tant els principals com els secundaris, ja que s'ha après a fer feina amb les tecnologies indicades i també s'ha aconseguit una presentació amigable, multiplataforma, intuïtiva i amb aparença nativa. Encara més, s'havia posat com a objectiu fer feina amb Java 17 i Spring Boot 2, i finalment s'ha fet amb Java 21 i Spring Boot 3, on aquest darrer introdueix canvis al mòdul de seguretat. També, tots els requeriments s'han satisfet, tant funcionals com no funcionals.

Per la part de planificació, ha estat bastant semblant a la realitat. Hauria estat bé fer ús d'una eina tipus Jira per poder haver tingut uns resultats més fidedignes i així haver pogut fer una gràfica de *burnout* [15] i veure el desviament de la planificació amb la realitat.

Per la part de desenvolupament, he trobat que el temps útil del qual disposem per aquesta tasca és poc. Si bé és important la documentació i la planificació, també ho és el desenvolupament, ja que realment és el que veu l'usuari. També entenc el temps reduït, perquè al final el temps és el que és i s'ha de fer encaixar el treball en un semestre.

Per la part de l'impacte ètic-socials, de sostenibilitat i diversitat, trobo que s'han aconseguit, pel fet que hem emprat tecnologies punteres que milloren l'empremta verda, i gràcies a la finalitat de l'aplicació, aconseguim reduir paper de manera eficaç.

Finalment, a títol personal, m'ha agradat molt conèixer Ionic Framework, i sobretot la seva adaptabilitat de disseny segons el dispositiu, i també estic molt satisfet amb el resultat final de l'aplicació, perquè la meua dona l'està utilitzant per estudiar les seves oposicions i veig que, sense haver d'explicar-li res, la sap fer servir.

5.2. Treball futur

- **Fer l'aplicació multiidioma.** Per poder arribar a més gent, cal implementar mecanismes d'internacionalització per permetre que l'aplicació es trobi en diferents idiomes. També, cal revisar els diferents formats de dates i números segons l'idioma.
- **Corregir el bug de l'expiració del token.** No és concebible que una aplicació et demani que iniciïs sessió cada dia. Això podria posar en perill l'èxit de l'aplicació. Per tant, cal dedicar temps a esbrinar la causa i corregir-la.
- **Implementar un mecanisme de restauració de contrasenya.** Si un usuari s'oblida de la seva contrasenya, ara per ara no té cap mecanisme de recuperació. Per tant, cal implementar-ho per assegurar que no hi hagi cap usuari que es quedi fora poder accedir-hi.
- **Millorar el registre quan l'usuari ja existeix.** S'ha d'informar adequadament a l'usuari de l'error quan intenta enregistrar-se amb dades d'un usuari existent, evitant obtenir l'error genèric.
- **Implementar autenticació amb OAuth de Google/Apple.** Avui dia molta gent no vol tenir un usuari per una aplicació concreta, s'estima més autenticar-se amb els seus comptes de Google o Apple. Per tant, per millorar l'experiència d'usuari, cal implementar aquesta integració.
- **Aplicar patró Redux per minimitzar les crides al backend.** Així com s'ha desenvolupat, les dades són en temps real. És a dir, es consulta per cada pantalla la informació al *backend*. Això pot suposar una càrrega important de rendiment si hi ha molts d'usuaris utilitzant-la. Amb el patró *Redux* podem minimitzar aquestes consultes gràcies a tenir les dades en memòria.
- **Transformar a PWA.** D'aquesta manera permetrem a l'usuari poder afegir l'aplicació web com si es tractés d'una aplicació del sistema. Així, l'usuari ja no cal que obri un navegador web i entri a l'aplicació.
- **Implementar nous tipus de test.** Actualment, l'aplicació només suporta exàmens de tipus test de resposta única, però per potenciar més l'eina, es podrien introduir altres modalitats, com la resposta múltiple.
- **Testejar l'aplicació.** És important que qualsevol aplicació disposi tant de proves unitàries com d'integració, de manera que podem assegurar que l'aplicació funciona tal com esperem. També, ens protegeix davant canvis, ja que en introduir nova funcionalitat podem rompre la ja existent i, sense tests, no ens n'adonàriem.

6. Glossari

A

Angular

Framework per aplicacions web desenvolupades en TypeScripti, ii, 4, 5, 7, 31, 42, 44, 51

B

backend

Arquitectura interna d'un lloc web i, ii, 5, 7, 10, 13, 31, 42, 43, 46, 47, 49, 50, 51, 53

bug

Malfuncionament del comportament de l'aplicació 48, 52

C

Capacitor Storage

API robusta per emmagatzemar data lleugera 31, 42, 47

compose

Eina per definir i executar aplicacions multicontenedor43, 48, 49, 50

D

DCU

Disseny Centrat en l'Usuari..... 11, 13, 29

Docker

Plataforma per construir, compartir i executar aplicacions43, 48, 49, 50

F

Flashcard

Targeta per estudiar i millorar la memòria amb l'associació ràpida d'informació3

Framework

Conjunt de biblioteques, estàndards i pautes 5, 7, 31, 42, 51, 52

frontend

Arquitectura d'un lloc web amb la que interactua l'usuari i, ii, 5, 7, 10, 13, 31, 42, 43, 47, 49, 50, 51

G

Gamificació

metodologia que té com a objectiu motivar mitjançant l'ús d'elements de joc.....2

H

Hibernate

Eina de mapeig relacional per Java..... 31, 49

HTTPS

Hypertext transfer protocol secure5

I

Ionic

Llibreria d'interfície d'usuari per crear aplicacions multiplataforma a partir d'un únic codi font i, ii, 4, 5, 7, 15, 31, 42, 44, 49, 51, 52

J

Jira

Aplicació per gestió de projectes i incidències52

JPA

Java Persistence API43

JWT

JSON Web Token31, 42, 43, 46, 47, 51

L

login

Procés d'identificació de l'usuari 10, 19, 31, 34, 44, 46, 47

M

MariaDB

Sistema gestor de base de dades derivat de MySQL31, 43

mockups

Prototip que mostra el funcionament d'una aplicació19, 24, 25, 26, 27, 30

multiplataforma

Que pot ser executat en una varietat de dispositius sense una adaptació específica.....1, 2, 3, 14, 15, 19, 52

O

OAuth

Open Authentication53

ODS

Objectius de Desenvolupament Sostenible5

OpenAPI

Llenguatge d'especificació d'API HTTP43

ORM

Object Relational Mapping.....31

P

PWA

Progressive Web Application53

R

Redux

Llibreria Javascript per gestionar l'estat53

REST

REpresentational State Transfer.....43, 44

S

Spring

Marc de desenvolupament d'aplicacions i contenidor d'inversió de control. i, ii, 4, 5, 7, 31, 42, 43, 46, 47, 51, 52

Spring Boot

Eina que simplifica el desenvolupament d'aplicacions amb Spring Framework..... i, ii, 4, 5, 7, 42, 52

Spring Security

Mòdul d'Spring personalitzable per l'autenticació i control d'accés..... 42, 46, 51

stack

Conjunt de tecnologies utilitzades43

T

TFM

Treball de Fi de Màster1, 4, 6, 7, 10, 11, 30

Thymeleaf

Llibreria Java que implementa un motor de plantilles d'XML/XHTML/5. 4, 42

token

Identificador únic d'accés vàlid d'un usuari..... 5, 31, 32, 42, 47, 49, 51, 52

Tomcat

Servidor HTTP de l'Apache Foundation.....31

U

UIB

Universitat de les Illes Balears1

W

wireframes

Esbós de l'esquelet d'un projectev, 19, 20, 21, 22, 23, 24, 25

7. Bibliografia

- [1] Google, «Qué puedes hacer con Formularios: Centro de Aprendizaje de Google Workspace,» [En línea]. Available: <https://support.google.com/a/users/answer/9302965?hl=es>. [Últim accés: 9 Març 2024].
- [2] Kahoot!, «Kahoot! for schools How it works: Kahoot!,» [En línea]. Available: <https://kahoot.com/schools/how-it-works/>. [Últim accés: 10 Març 2024].
- [3] «Choose a Plan: Socrative,» [En línea]. Available: <https://www.socrative.com/plans>. [Últim accés: 9 Març 2024].
- [4] Quizlet, «About Quizlet: Quizlet,» [En línea]. Available: <https://quizlet.com/mission>. [Últim accés: 10 Març 2024].
- [5] ClassMarker, «Online exam maker features: ClassMarker,» [En línea]. Available: <https://www.classmarker.com/online-testing/quiz-features/>. [Últim accés: 10 Març 2024].
- [6] B. Buchanan, «Which Is One of The Most Wasteful Industries for Energy In The World ... Software!: Medium,» 24 Juliol 2024. [En línea]. Available: <https://medium.com/asecuritysite-when-bob-met-alice/which-is-one-of-the-most-wasteful-energy-industries-in-the-world-software-8d90247e9e79>. [Últim accés: 7 Març 2024].
- [7] ScrumGuides, «The 2020 Scrum Guide: Scrum Guides,» Novembre 2020. [En línea]. Available: <https://scrumguides.org/scrum-guide.html#scrum-team>. [Últim accés: 9 Març 2024].
- [8] Adobe, «Waterfall Methodology: A Complete Guide: Adobe Experience Cloud Blog,» 18 Març 2022. [En línea]. Available: <https://business.adobe.com/blog/basics/waterfall#disadvantages-of-the-waterfall-methodology>. [Últim accés: 8 Març 2024].
- [9] Kanban University, «The Kanban Method: Kanban University,» [En línea]. Available: <https://kanban.university/kanban-guide>. [Últim accés: 8 Març 2024].
- [10] Unir, «¿En qué consiste el diseño centrado en el usuario?,» 21 Desembre 2022. [En línea]. Available: <https://www.unir.net/marketing-comunicacion/revista/disenio-centrado-usuario/>. [Últim accés: 27 Març 2024].
- [11] BBC, «¿A qué edad alcanzamos el punto máximo de inteligencia?,» 16 Març 2019. [En línea]. Available: <https://www.bbc.com/mundo/noticias-47523412>. [Últim accés: 29 Març 2024].
- [12] Google, «Bottom navigation,» [En línea]. Available: <https://m2.material.io/components/bottom-navigation#usage>. [Últim accés: 28 Març 2024].
- [13] Universitat Oberta de Catalunya, «Test amb usuaris. Disseny d'interfícies.,» [En línea]. Available: <https://multimedia.uoc.edu/blogs/dii/disseny-centrat-en-lusuari-dcu/test-amb-usuaris/>. [Últim accés: 04 Abril 2024].
- [14] Capacitor, «Data Storage in Capacitor,» [En línea]. Available: <https://capacitorjs.com/docs/guides/storage>. [Últim accés: 06 Abril 2024].

- [15] Atlassian, «Jira Burndown Chart,» [En línia]. Available: <https://www.atlassian.com/agile/tutorials/burndown-charts>. [Últim accés: 03 Juny 2024].

8. Annexos

Els annexos es troben en documents separats, no integrats en la present memòria principal.

Enquestes dels usuaris	Annex I
Estadístiques de les enquestes	Annex II
Manual d'instal·lació	Annex III